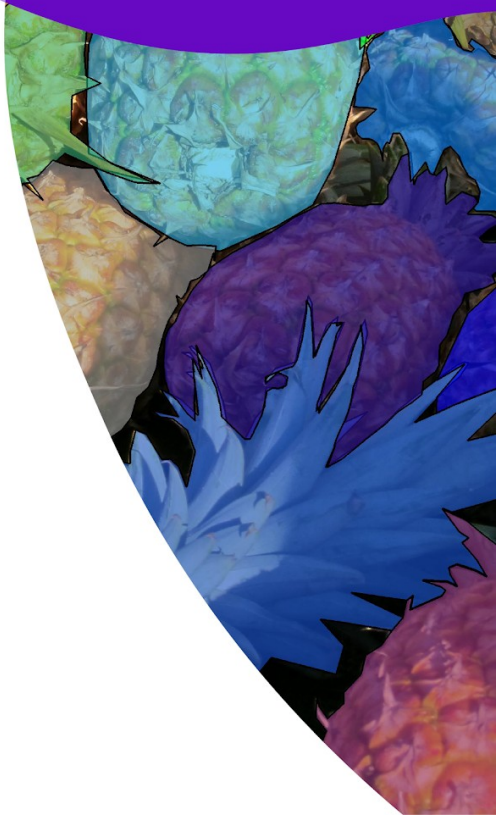
NEW YORK UNIVERSITY

Deep Learning

http://bit.ly/DLSP20



Yann LeCun
NYU - Courant Institute & Center for Data Science
Facebook AI Research
http://yann.lecun.com
TAs: Alfredo Canziani, Mark Goldstein

NYU DL Spring 2020

Course information

- **Website:**
  - http://bit.ly/DLSP20
- **TA: Alfredo Canziani & Mark Goldstein**
- **Lectures:**
  - 9 lectures by YLC
  - 3 guest lectures
- **Practical session**
  - Tuesday evenings with Alfredo
- **Evaluation**
  - Mid-term exam
  - Final project (on self-supervised learning & autonomous driving)

Course Plan (1/3)

- **Basics of Supervised Learning, Neural Nets, Deep Learning.**
  - What DL can do
  - What are good features / representations
- **Backpropropagation and architectural components**
  - Modules, gradients, Architectures, losses, activations
  - Weight sharing / tying, Multiplicative interactions / sum-product / attention / gating
  - Mixture of experts, Siamese nets, hyper-networks
- **Convnets & applications 1**
- **Convnets & applications 2**
- **More DL Architectures**
  - Recurrent nets, BPTT / applications, truck backer-upper
  - GRU / LSTM, Memory nets, Transformers / adapters
  - Graph NN

Course Plan (2/3)

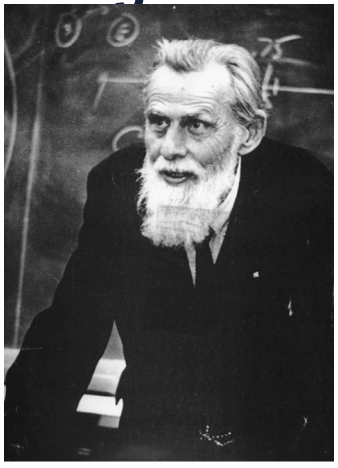
- **Regularization tricks / Optimization tricks / understanding how DL works**
  - Convergence of (convex) optimization
  - Geometry of the objective function
  - Initialization tricks, Normalization tricks, Drop out, gradient clipping...
  - Momentum, average SGD, Parallelization of SGD
  - Target prop, Lagrangian formulation
- **Energy-based models**
  - Notations, Latent variable models, latent variable inference & regularization
  - Minimization, marginalization, free energy
  - Structured prediction / Reasoning as energy minimization
  - Sparse modeling / k-means / PCA / Convolutional sparse coding

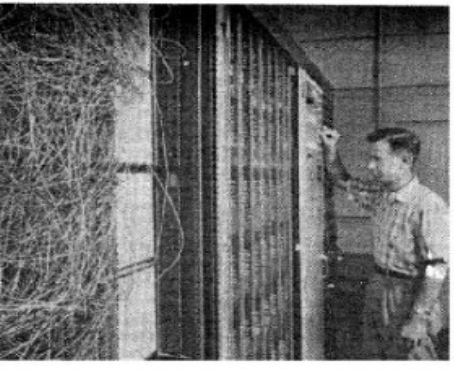
Course Plan (3/3)

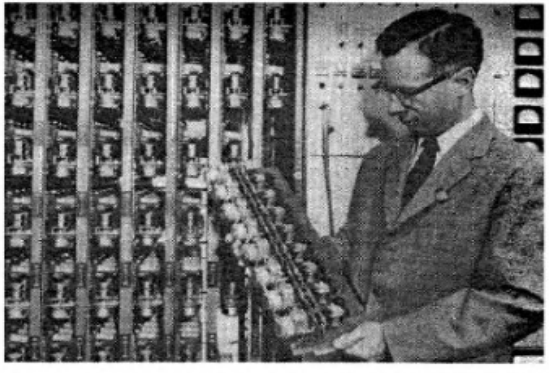
- **Self-supervised learning**
  - Contrastive methods and Regularization methods for energy shaping
  - Accelerated inference: encoder, LISTA, VAE
  - Denoising AE, variational AE, contrastive divergence….
  - Generative adversarial Networks
- **SSL and beyond**
  - How does human and animal learning work?
  - How do we get to human-level AI?
  - Building models of the world for control

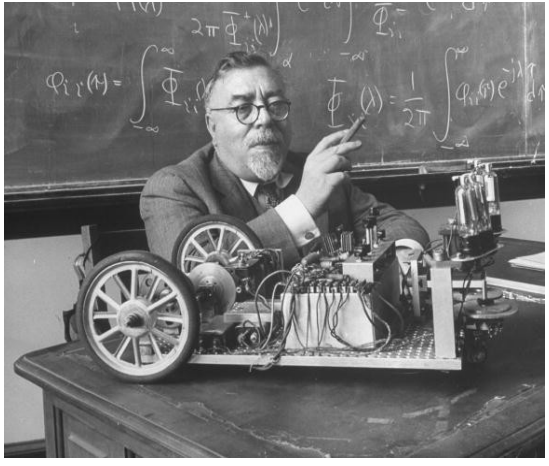
Inspiration for Deep Learning: The Brain!

- McCulloch & Pitts (1943): networks of binary neurons can do logic
- Donald Hebb (1947): Hebbian synaptic plasticity
- Norbert Wiener (1948): cybernetics, optimal filter, feedback, autopoïesis, auto-organization.
- Frank Rosenblatt (1957): Perceptron
- Hubel & Wiesel (1960s): visual cortex architecture



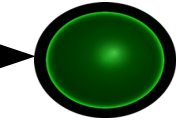


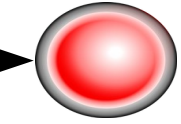




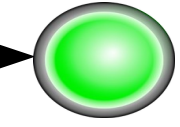
Supervised Learning

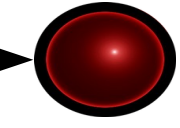
- **Training a machine by showing examples instead of programming it**
- **When the output is wrong, tweak the parameters of the machine**
- **Works well for:**
  - Speech → words
  - Image → categories
  - Portrait → name
  - Photo → caption
  - Text → topic
  - ….





CAR



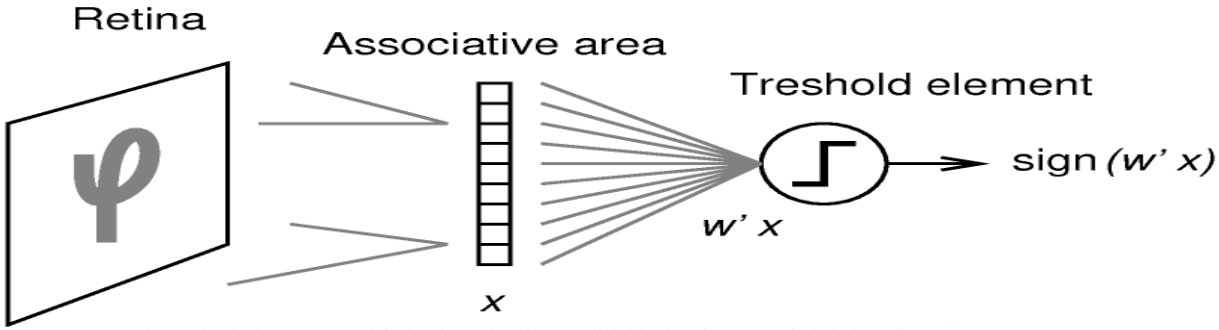


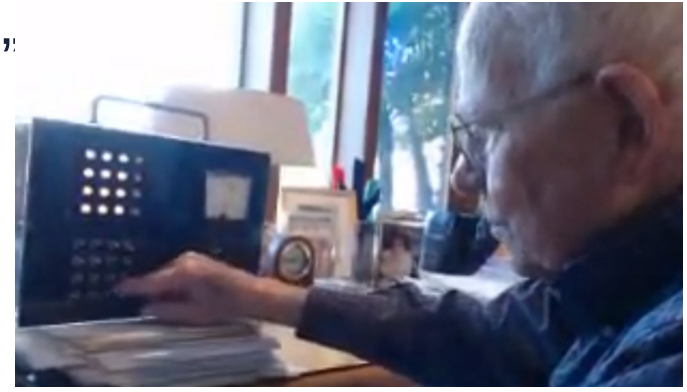
PLANE

Supervised Learning goes back to the Perceptron & Adaline

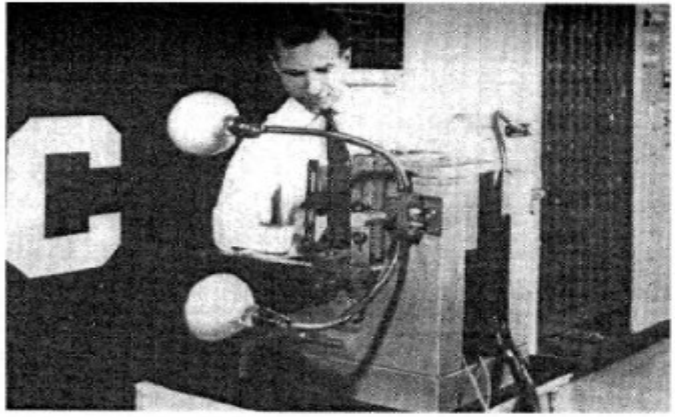
- **The McCulloch-Pitts Binary Neuron**
  - Perceptron: weights are motorized potentiometers
  - Adaline: Weights are electrochemical "memistors"

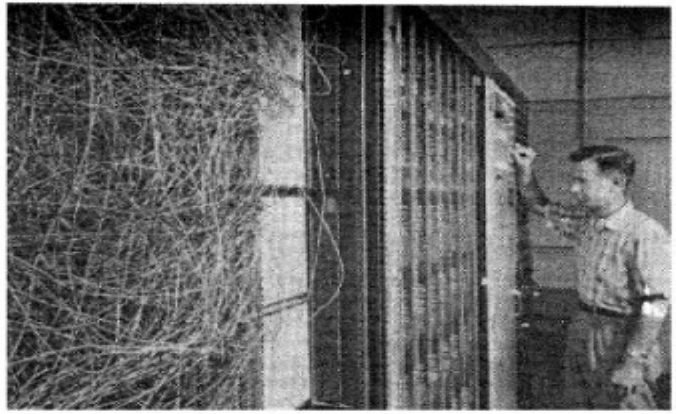
$$y = sign\left(\sum_{i=1}^{N} W_i X_i + b\right)$$

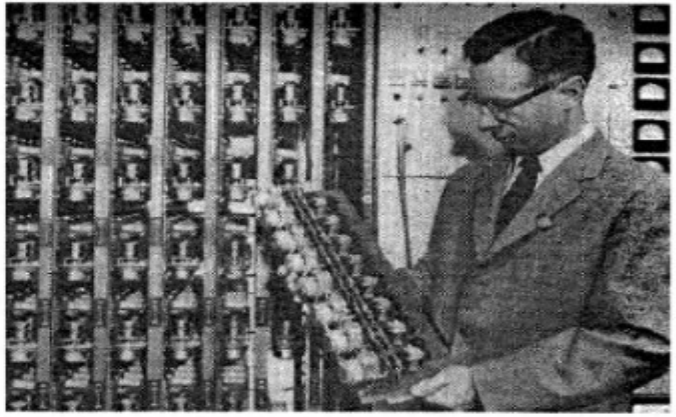




https://youtu.be/X1G2g3SiCwU

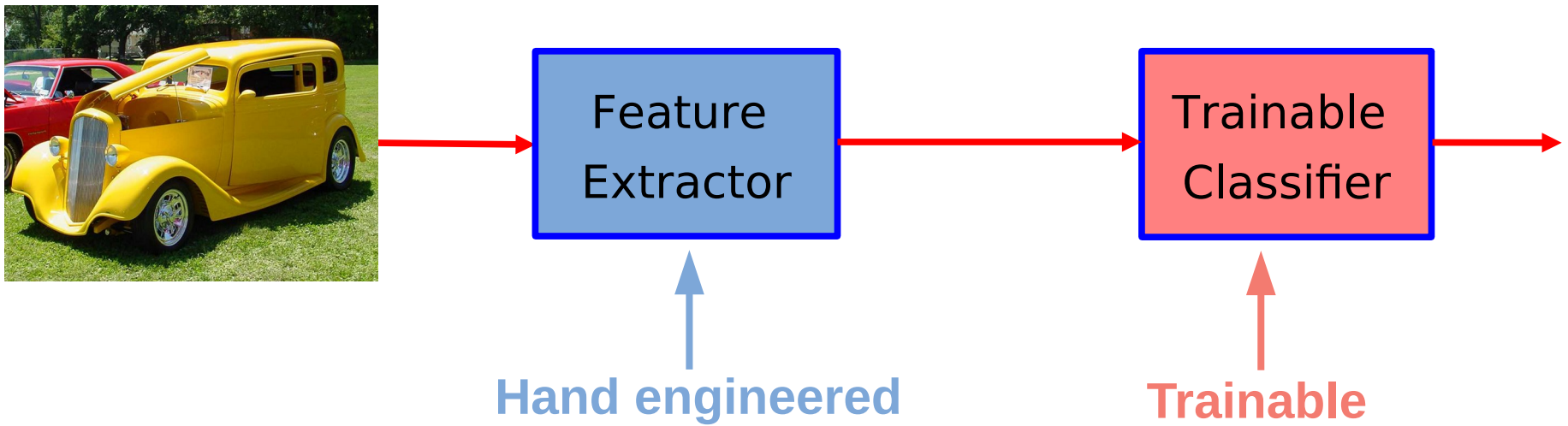






The Standard Paradigm of Pattern Recognition

- **...and “traditional” Machine Learning**



Multilayer Neural Nets and Deep Learning

Traditional Machine Learning

Feature Extractor

Trainable Classifier

Hand engineered

Trainable

Deep Learning

Trainable

Low-Level Features

Mid-Level Features

High-Level Features

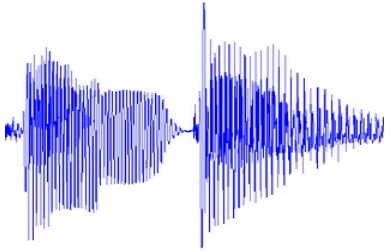
Trainable Classifier

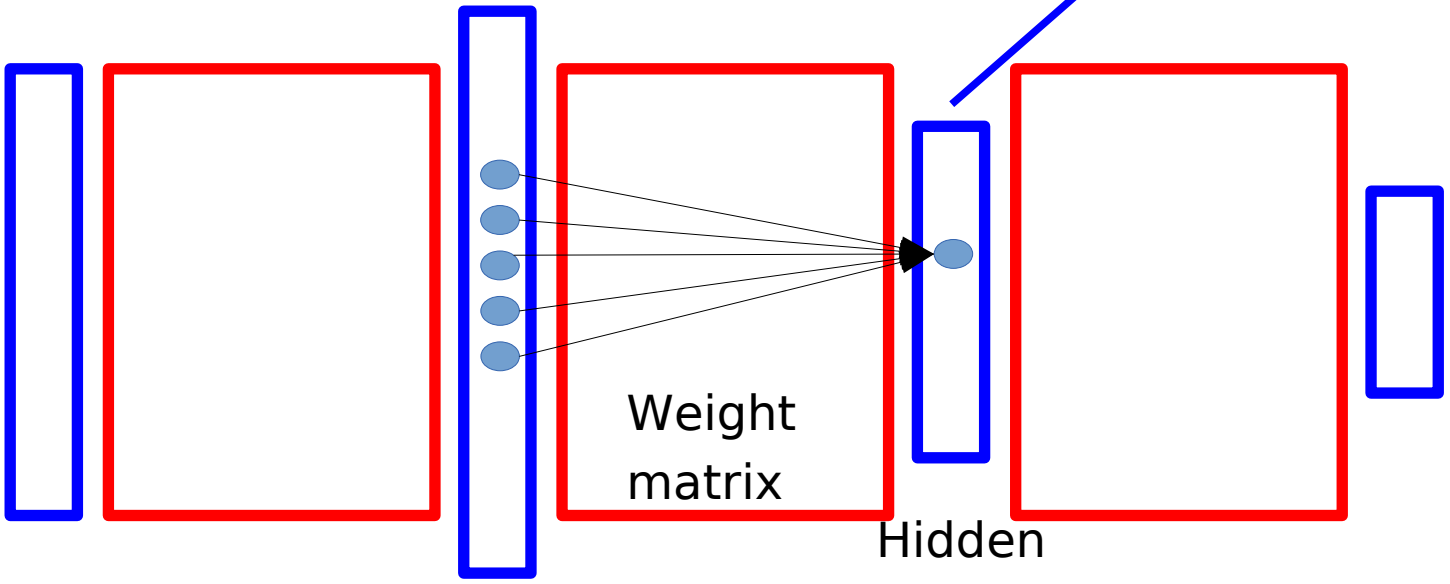
(Deep) Multi-Layer Neural Nets

- Multiple Layers of simple units
- Each units computes a weighted sum of its inputs
- Weighted sum is passed through a non-linear function
- The learning algorithm changes the weights

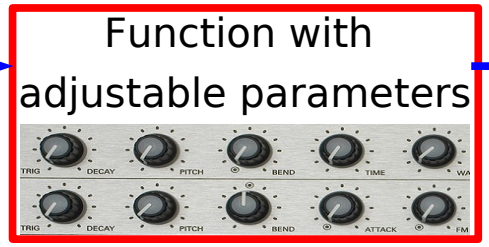
$$ReLU(x) = max(x, 0)$$

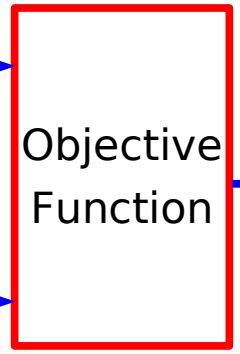
Ceci est une voiture



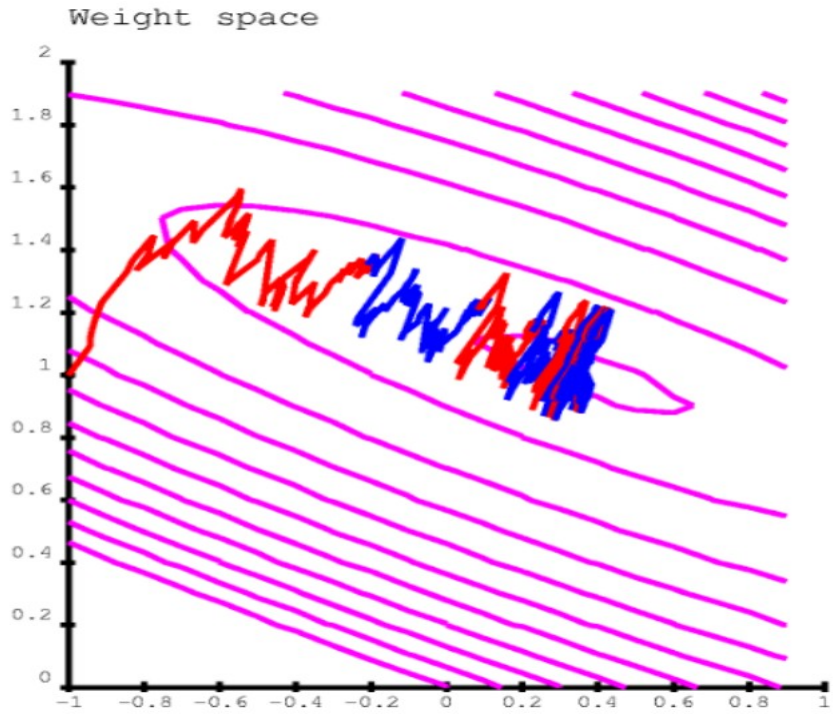


Supervised Machine Learning = Function Optimization





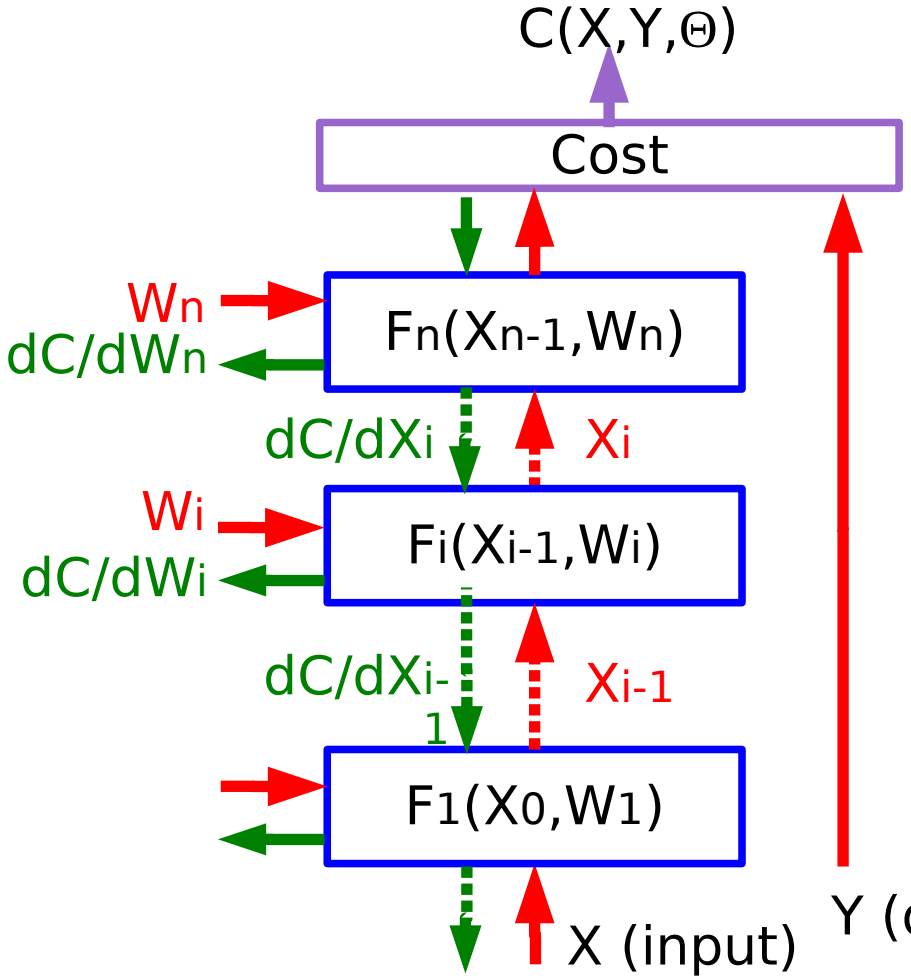
traffic light: -1



- **It's like walking in the mountains in a fog and following the direction of steepest descent to reach the village in the valley**
- **But each sample gives us a noisy estimate of the direction. So our path is a bit random.**
- **Stochastic Gradient Descent (SGD)**

$$W_i \leftarrow W_i - \eta \frac{\partial L(W, X)}{\partial W_i}$$

Computing Gradients by Back-Propagation



- **A practical Application of Chain Rule**

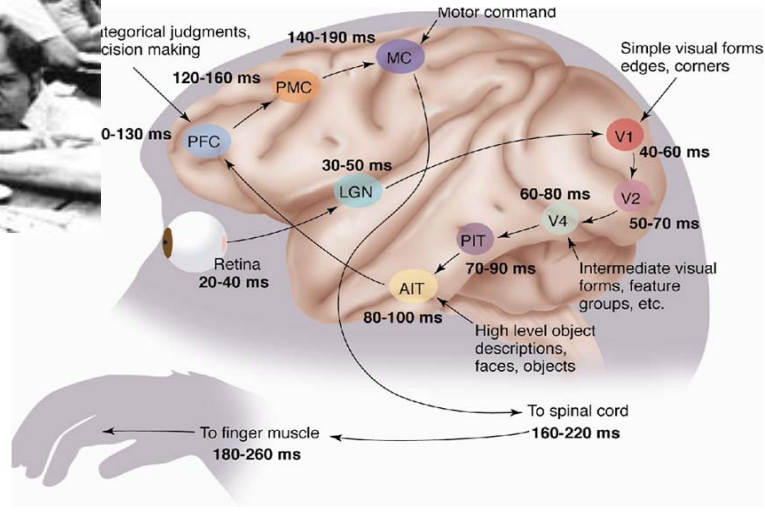
- **Backprop for the state gradients:**
- **$dC/dX_{i-1} = dC/dX_i \cdot dX_i/dX_{i-1}$**
- **$dC/dX_{i-1} = dC/dX_i \cdot dF_i(X_{i-1},W_i)/dX_{i-1}$**

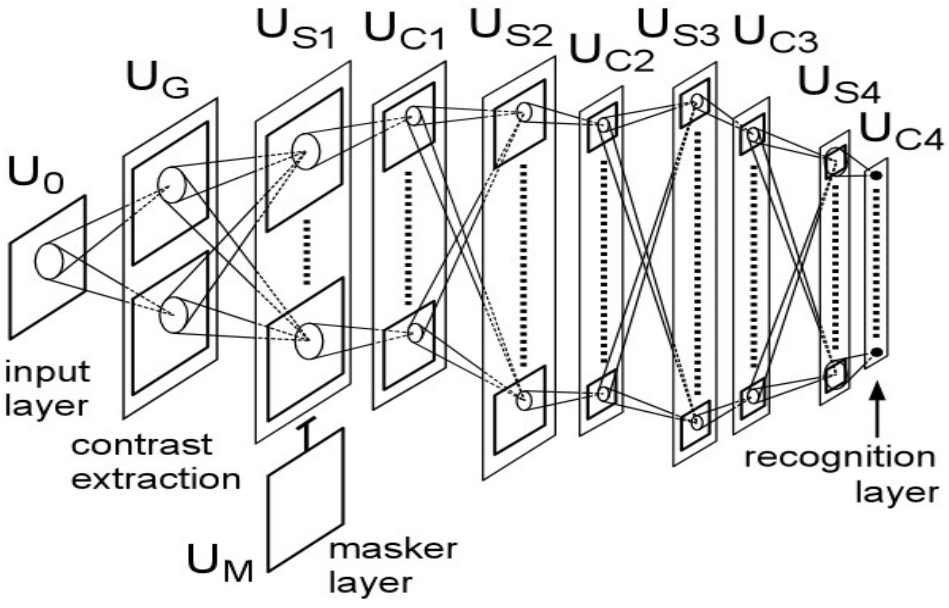
- **Backprop for the weight gradients:**
- **$dC/dW_i = dC/dX_i \cdot dX_i/dW_i$**
- **$dC/dW_i = dC/dX_i \cdot dF_i(X_{i-1},W_i)/dW_i$**

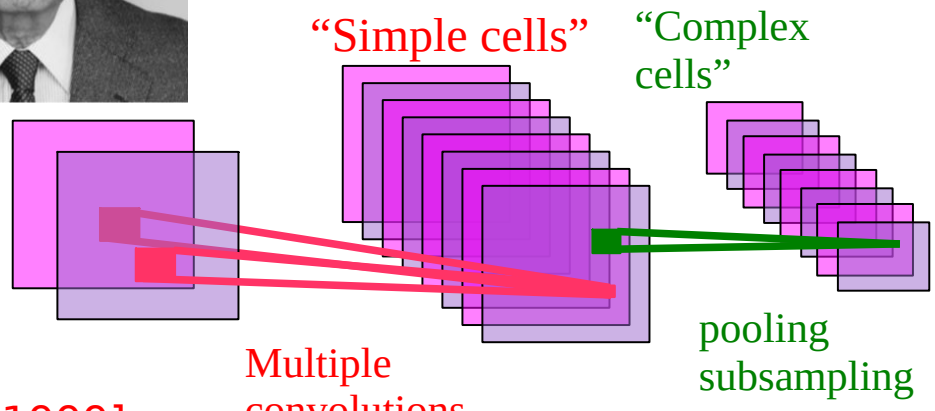
Hubel & Wiesel's Model of the Architecture of the Visual Cortex

- **[Hubel & Wiesel 1962]:**
  - simple cells detect local features
  - complex cells "pool" the outputs of simple cells within a

[Thorpe & Fabre-Thorpe 2001]

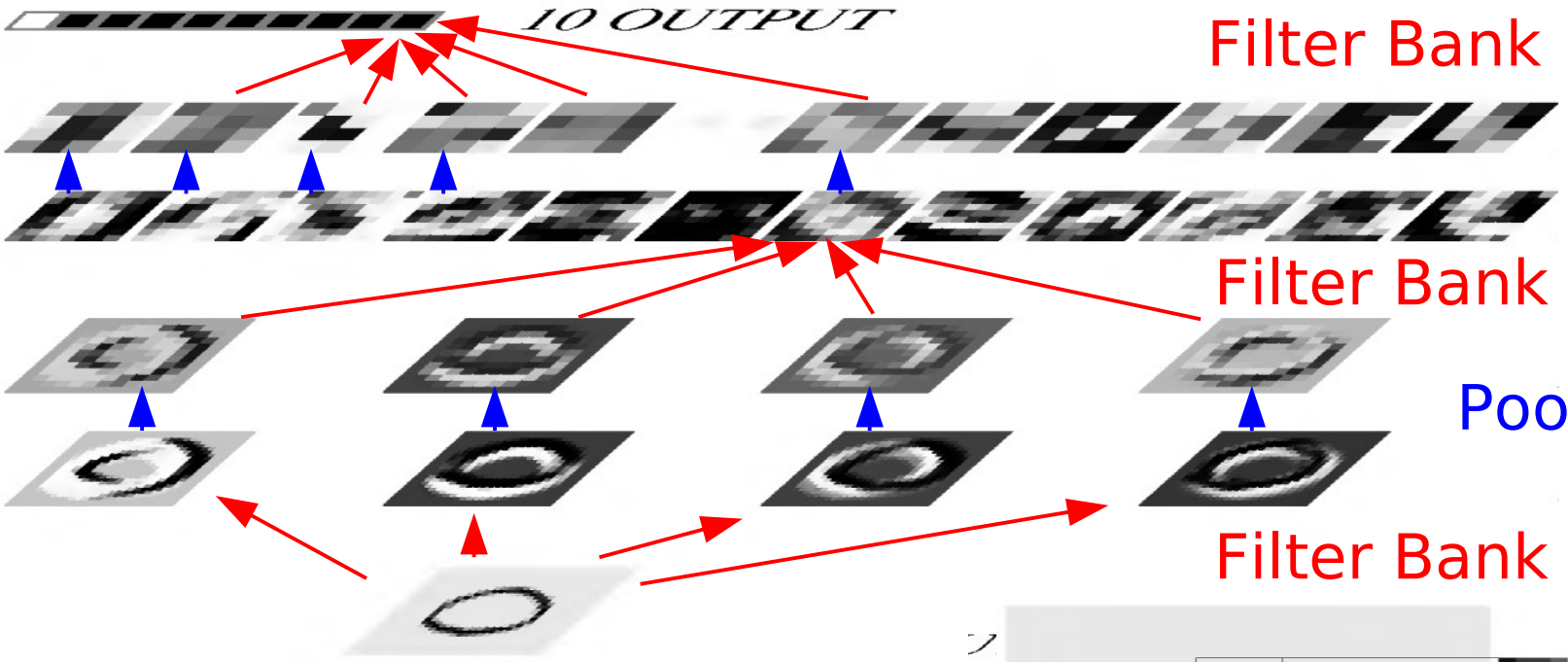






[Fukushima 1982][LeCun 1989, 1998],[Riesenhuber 1999].....

Convolutional Network Architecture [LeCun et al. NIPS 1989]



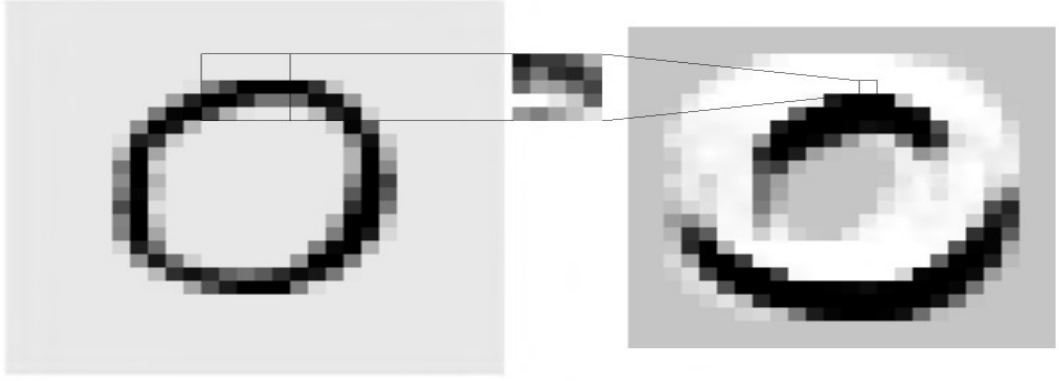
Filter Bank +non-linearity

Pooling

Filter Bank +non-linearity

Pooling

Filter Bank +non-linearity



- **Inspired by [Hubel & Wiesel 1962] & [Fukushima 1982] (Neocognitron):**
  - simple cells detect local features
  - complex cells “pool” the outputs of simple cells within a retinotopic neighborhood.

Convolutional Network (LeNet5, vintage 1990)

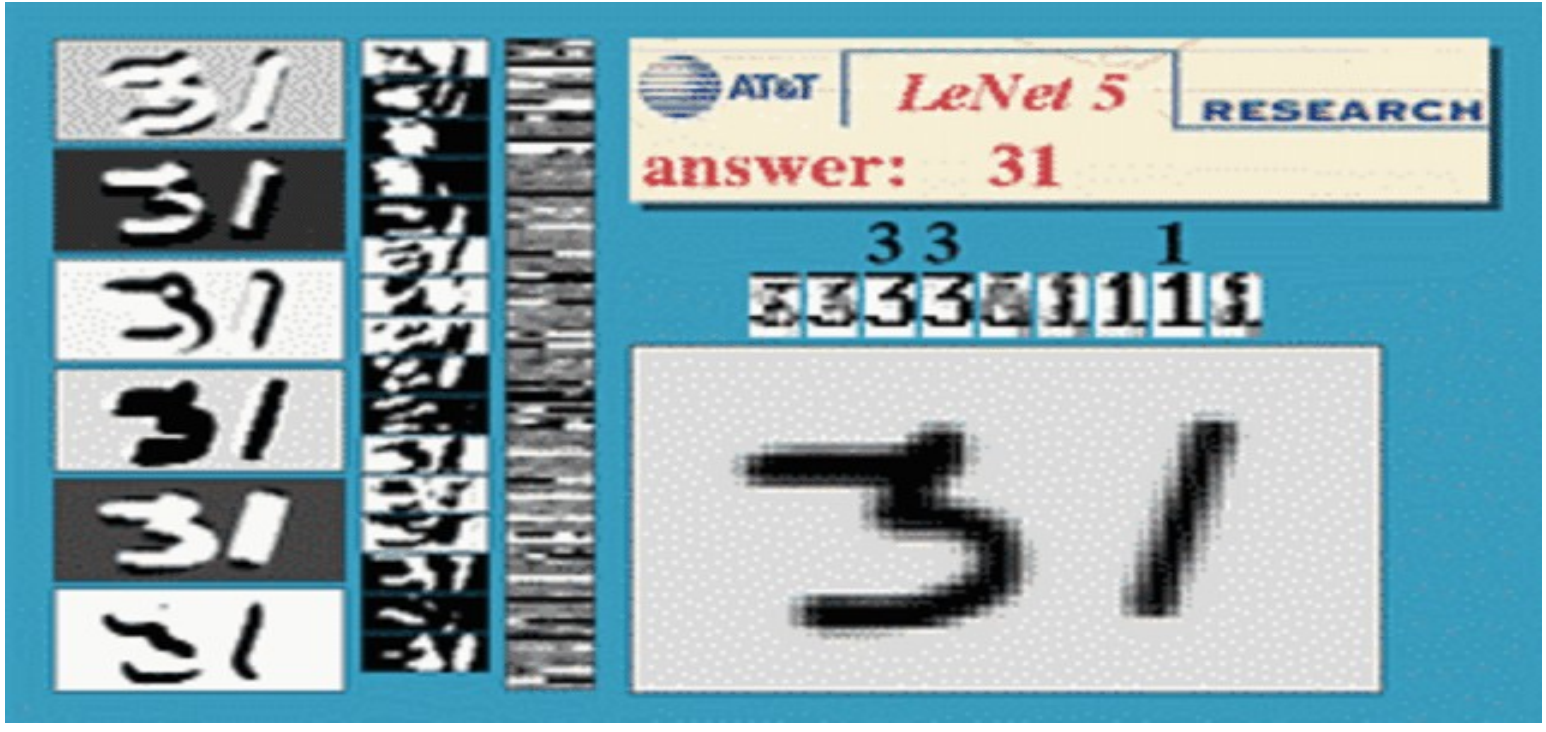
- **Filters-tanh → pooling → filters-tanh → pooling → filters-tanh**

LeNet character recognition demo 1992

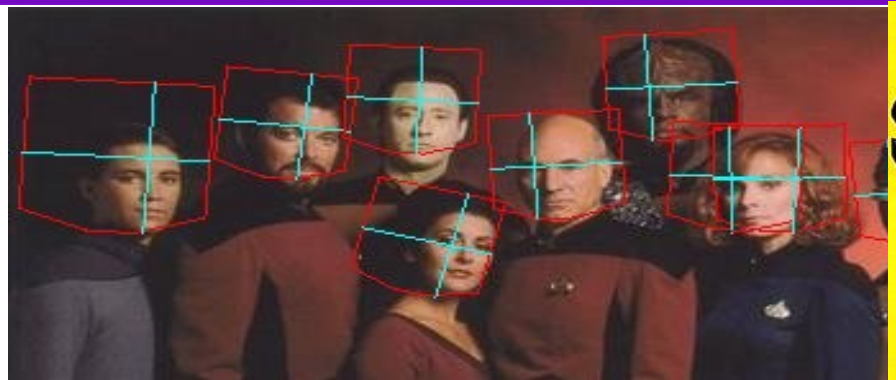
- **Running on an AT&T DSP32C (floating-point DSP, 20 MFLOPS)**

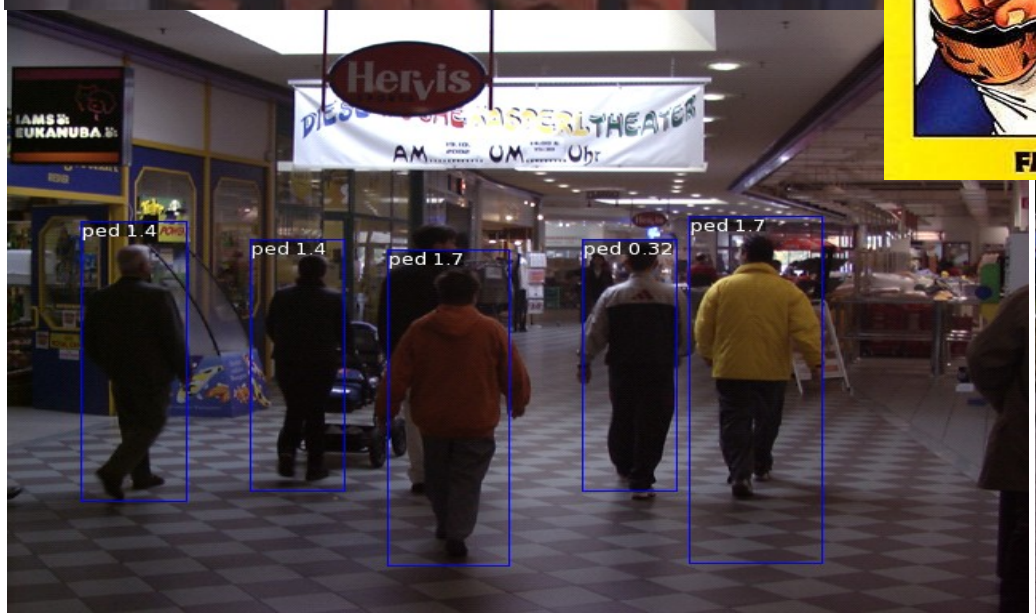
ConvNets can recognize multiple objects

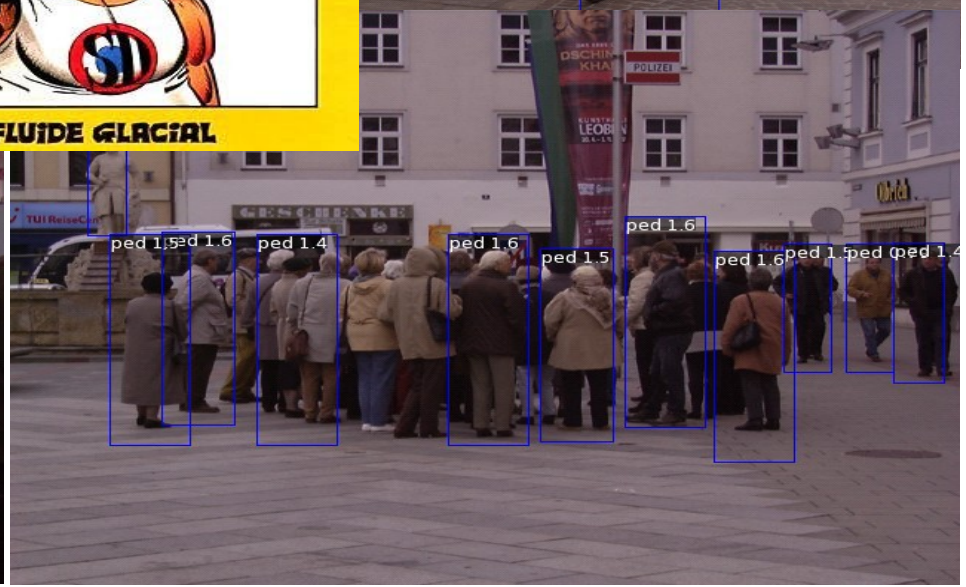
- **All layers are convolutional**
- **Networks performs simultaneous segmentation and recognition**
  - [LeCun, Bottou, Bengio, Haffner, Proc IEEE 1998]



Face & Pedestrian Detection with ConvNets (1993-2005)

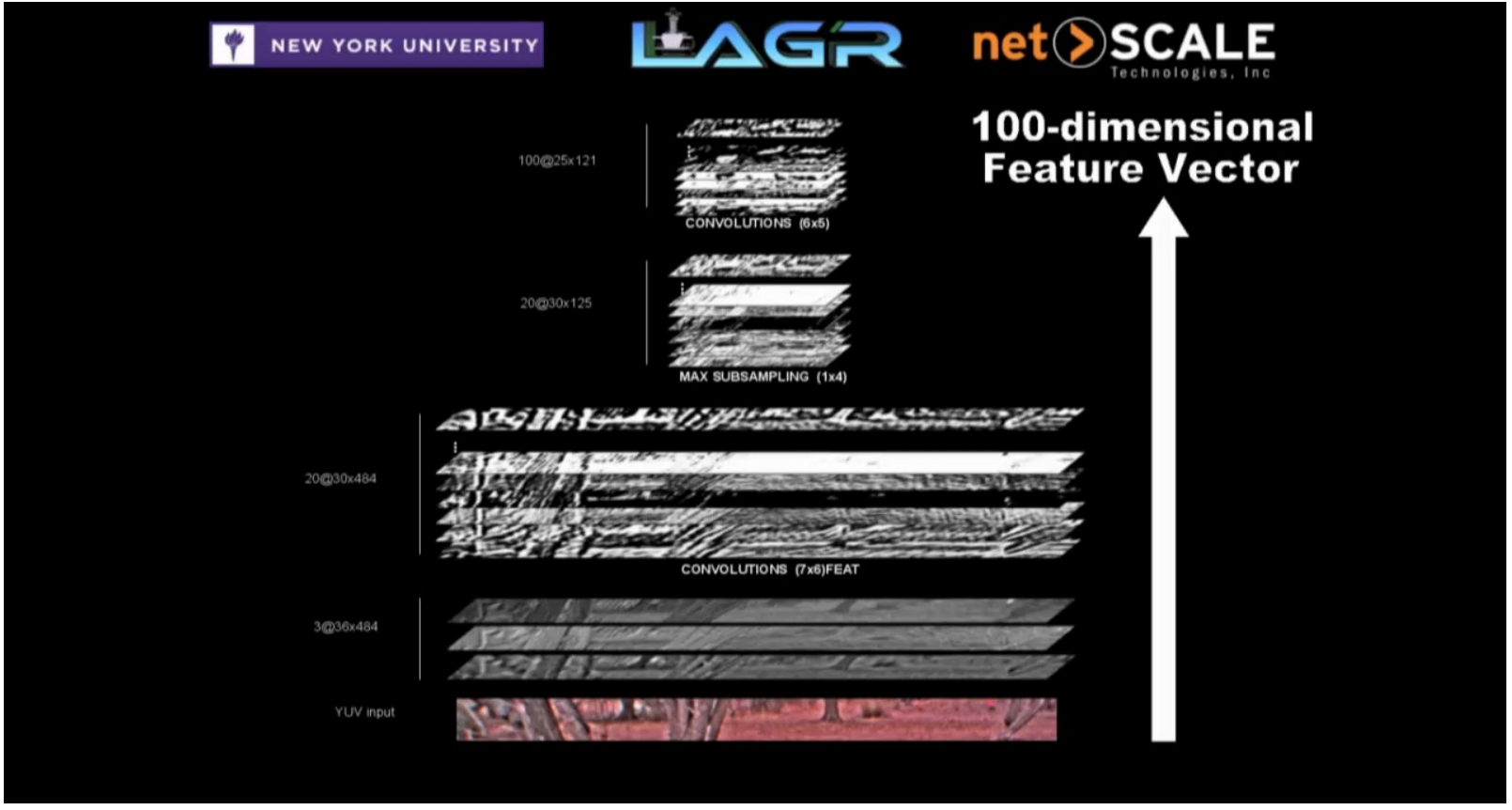






[Osadchy,Miller LeCun JMLR 2007],[Kavukcuoglu et al. NIPS 2010] [Sermanet et al. CVPR 2013]

Training a Robot to Drive Itself in Nature [Hadsell 2009]

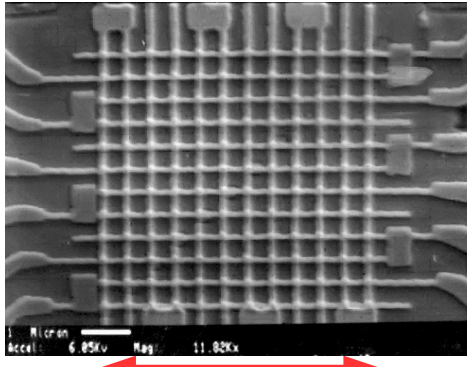


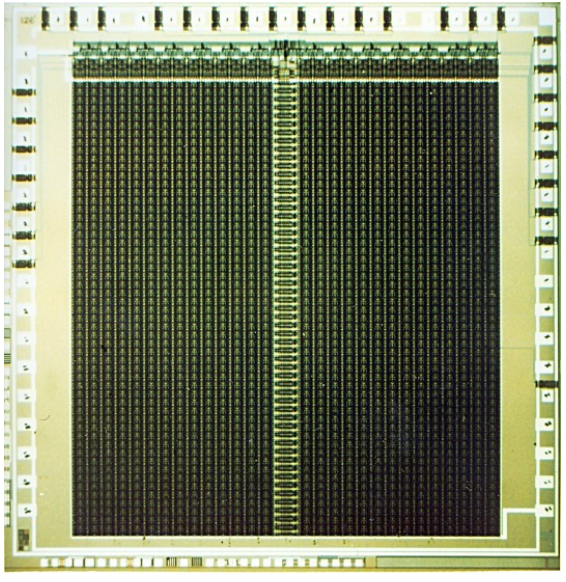
Semantic Segmentation with ConvNets [Farabet 2012]

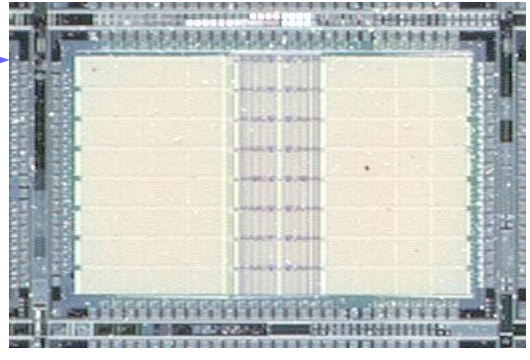
- **33 categories**

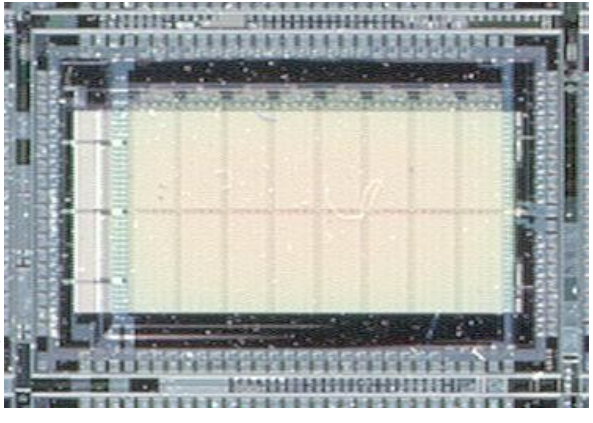
1986-1996 Neural Net Hardware at Bell Labs, Holmdel

- **1986: 12x12 resistor array**
  - Fixed resistor values
  - E-beam lithography: 6x6microns
- **1988: 54x54 neural net**
  - Programmable ternary weights
  - On-chip amplifiers and I/O
- **1991: Net32k: 256x128 net**
  - Programmable ternary weights
  - 320GOPS, 1-bit convolver.
- **1992: ANNA: 64x64 net**
  - ConvNet accelerator: 4GOPS
  - 6-bit weights, 3-bit activations



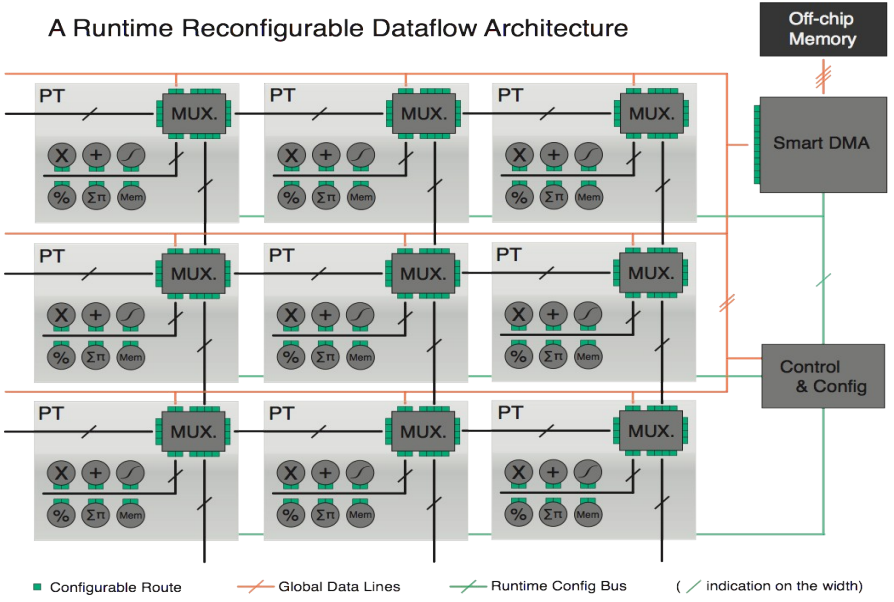




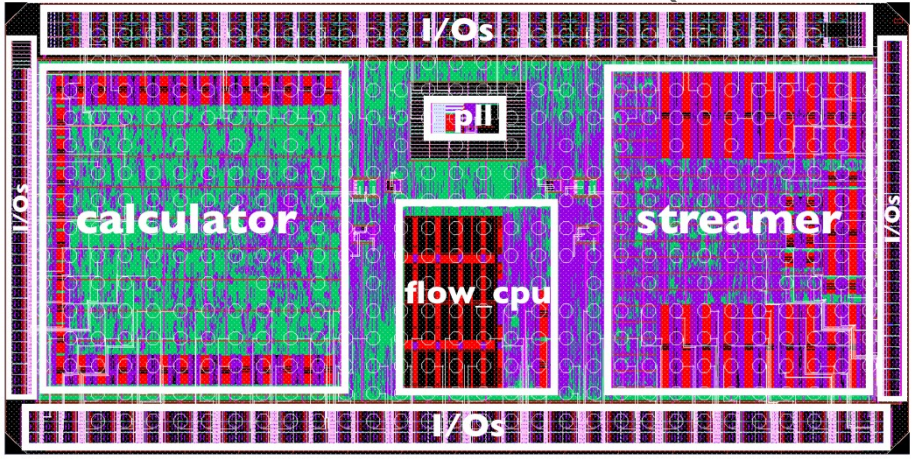


FPGA ConvNet Accelerator: NewFlow [Farabet 2011]

- **NeuFlow: Reconfigurable Dataflow architecture**
  - Implemented on Xilinx Virtex6 FPGA
  - 20 configurable tiles. 150GOPS, 10 Watts
  - Semantic Segmentation: 20 frames/sec at 320x240
  - **Exploits the structure of convolutions**

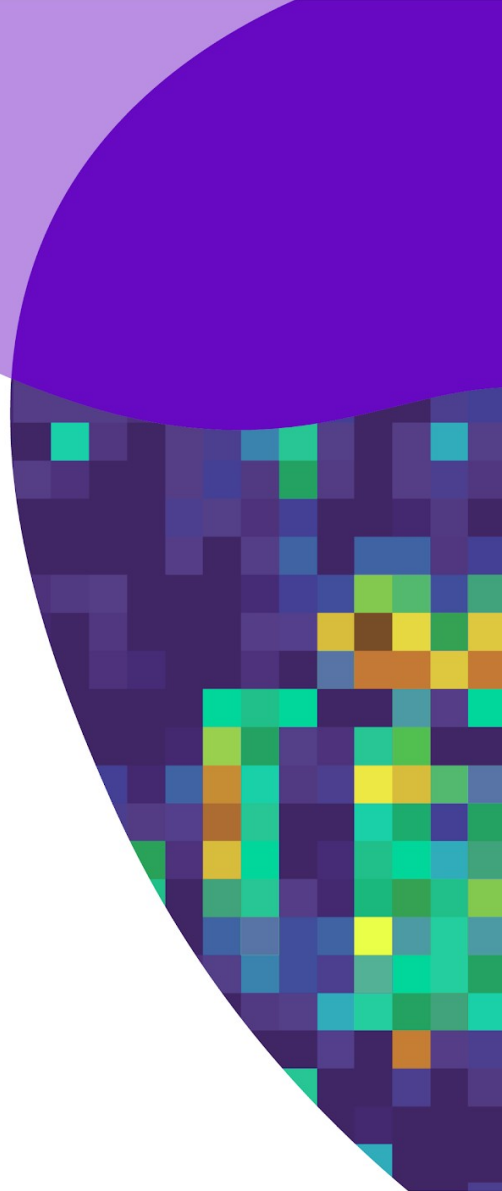


- **NeuFlow ASIC** [Pham 2012]
  - 150GOPS, 0.5 Watts (simulated)



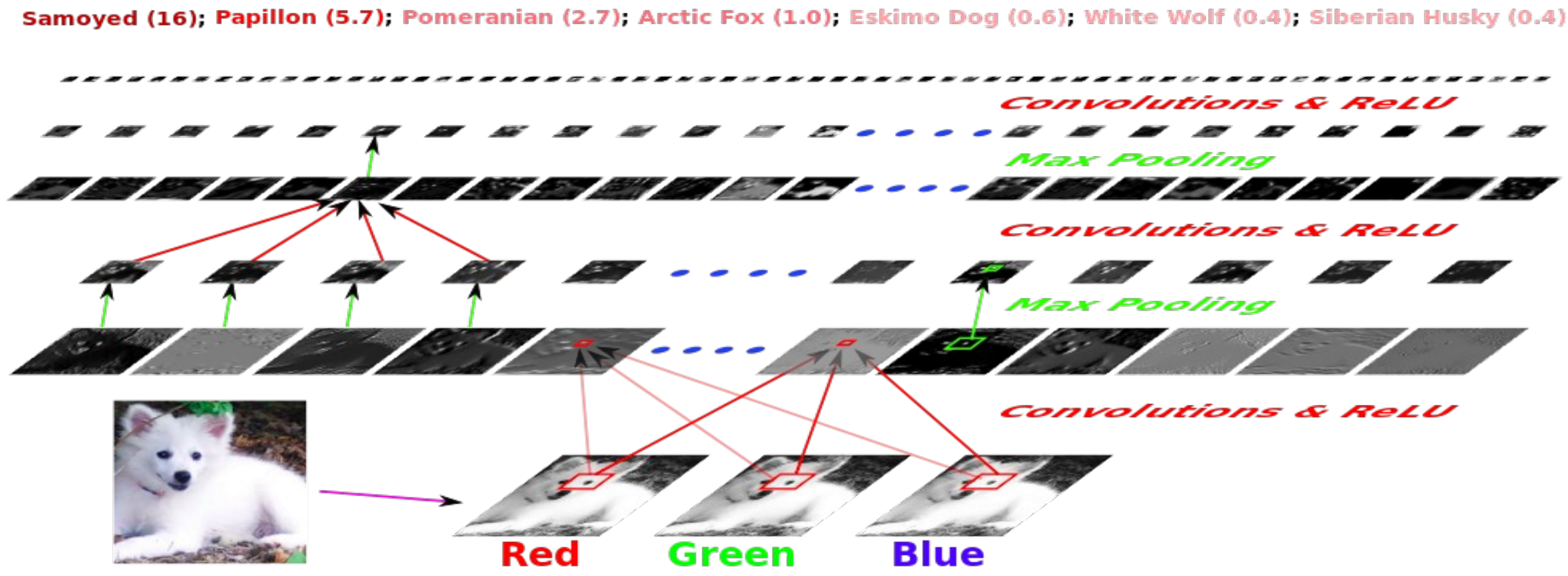
The Deep Learning Revolution

Speech recognition: 2010
Image recognition: 2013
Natural language processing: 2015



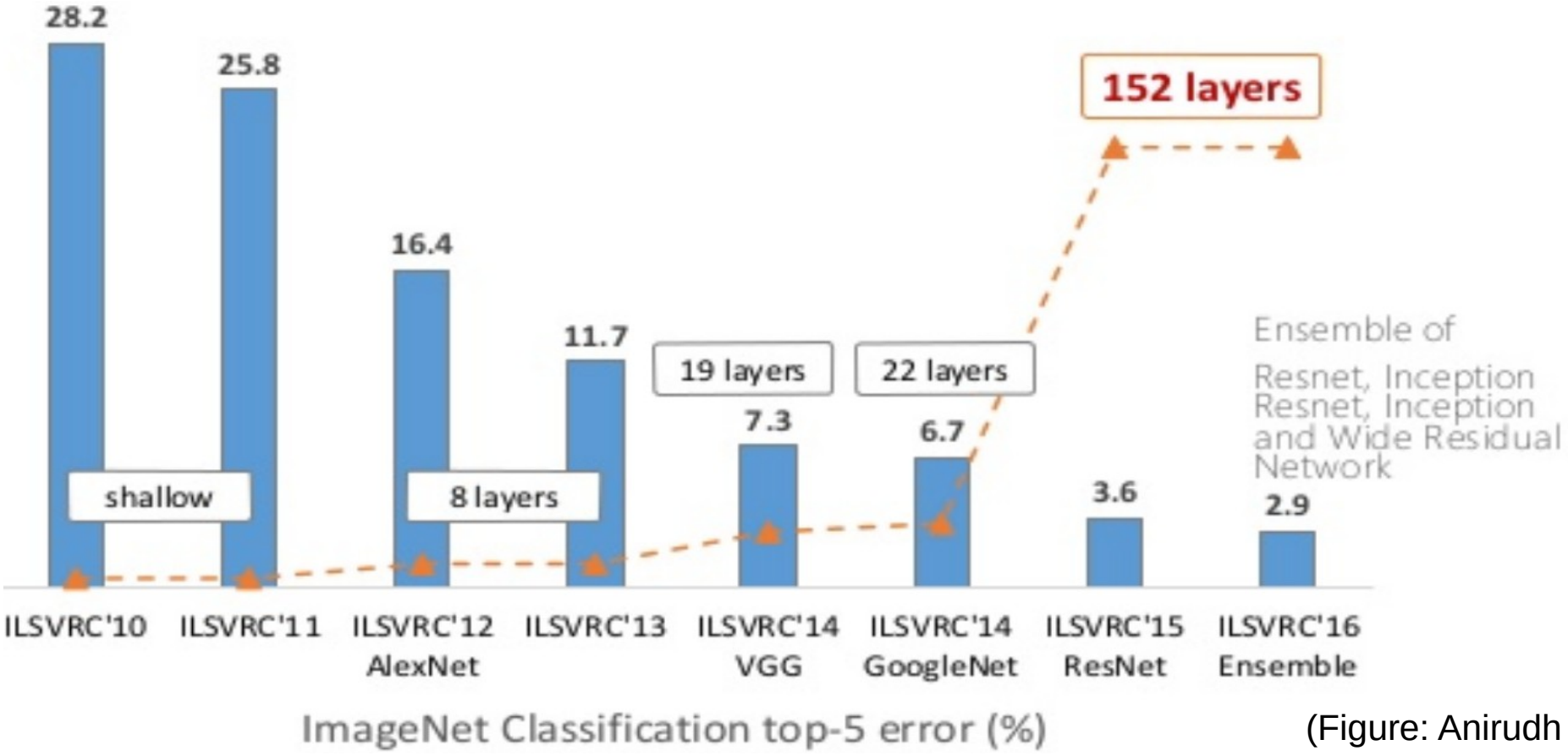
Deep ConvNets for Object Recognition (on GPU)

- AlexNet [Krizhevsky et al. NIPS 2012], OverFeat [Sermanet et al. 2013]
- 1 to 10 billion connections, 10 million to 1 billion parameters, 8 to 20 layers.



Error Rate on ImageNet

- **Depth inflation**

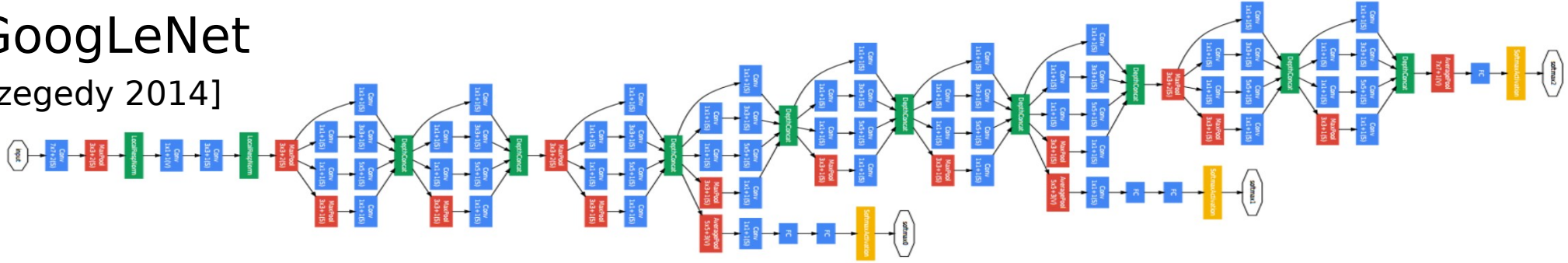


(Figure: Anirudh Koul)

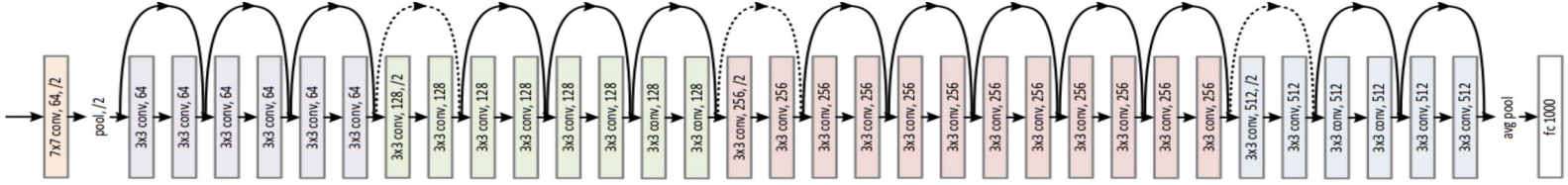
Deep ConvNets: depth inflation!

VGG
[Simonyan 2013]

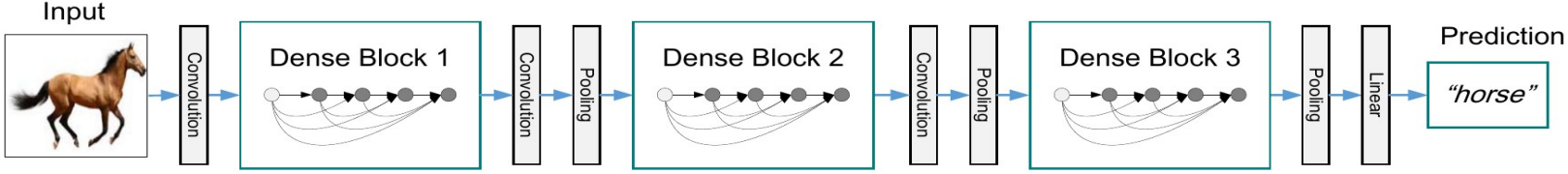
GoogLeNet
Szegedy 2014]



ResNet
[He et al. 2015]

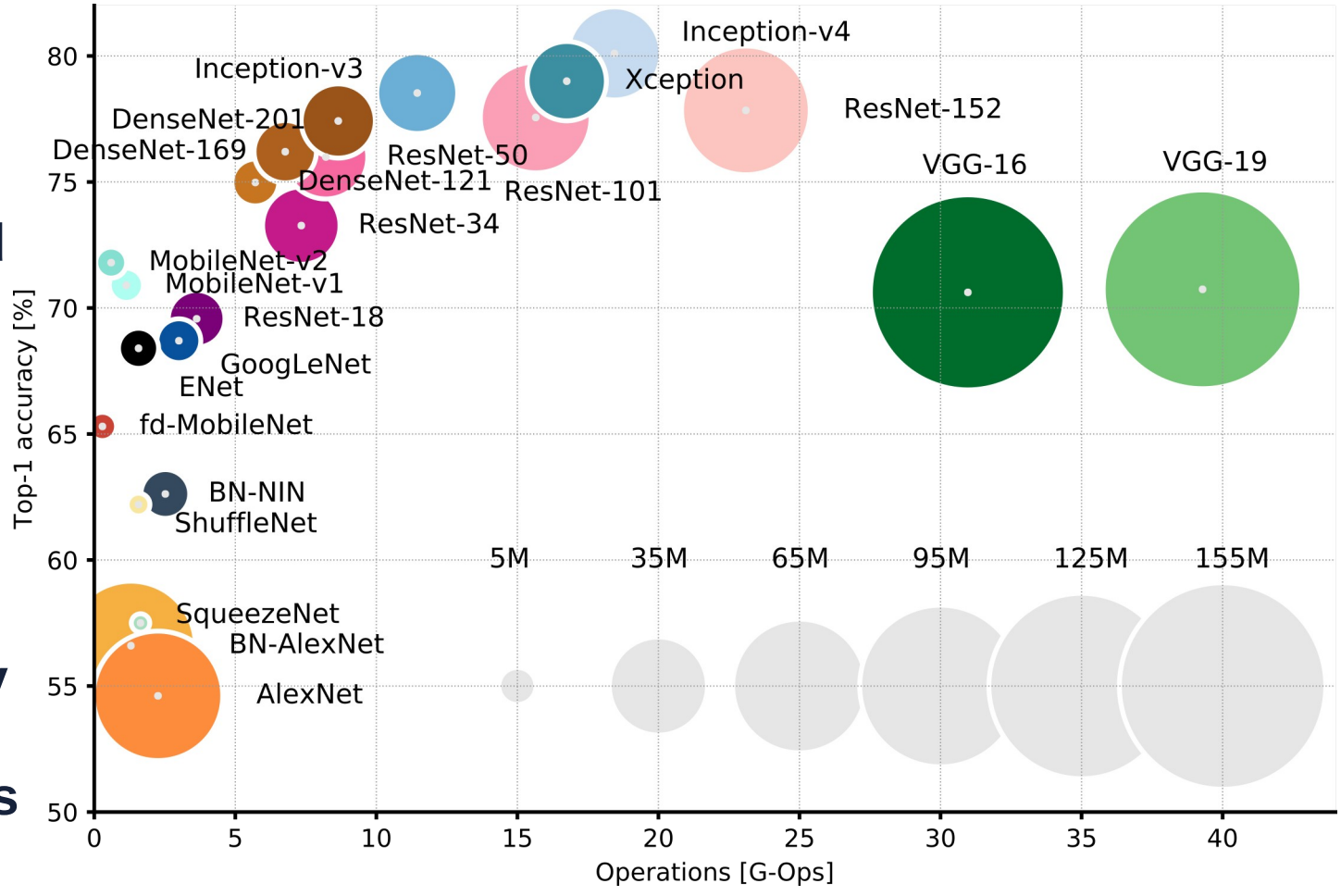


DenseNet
[Huang et al 2017]



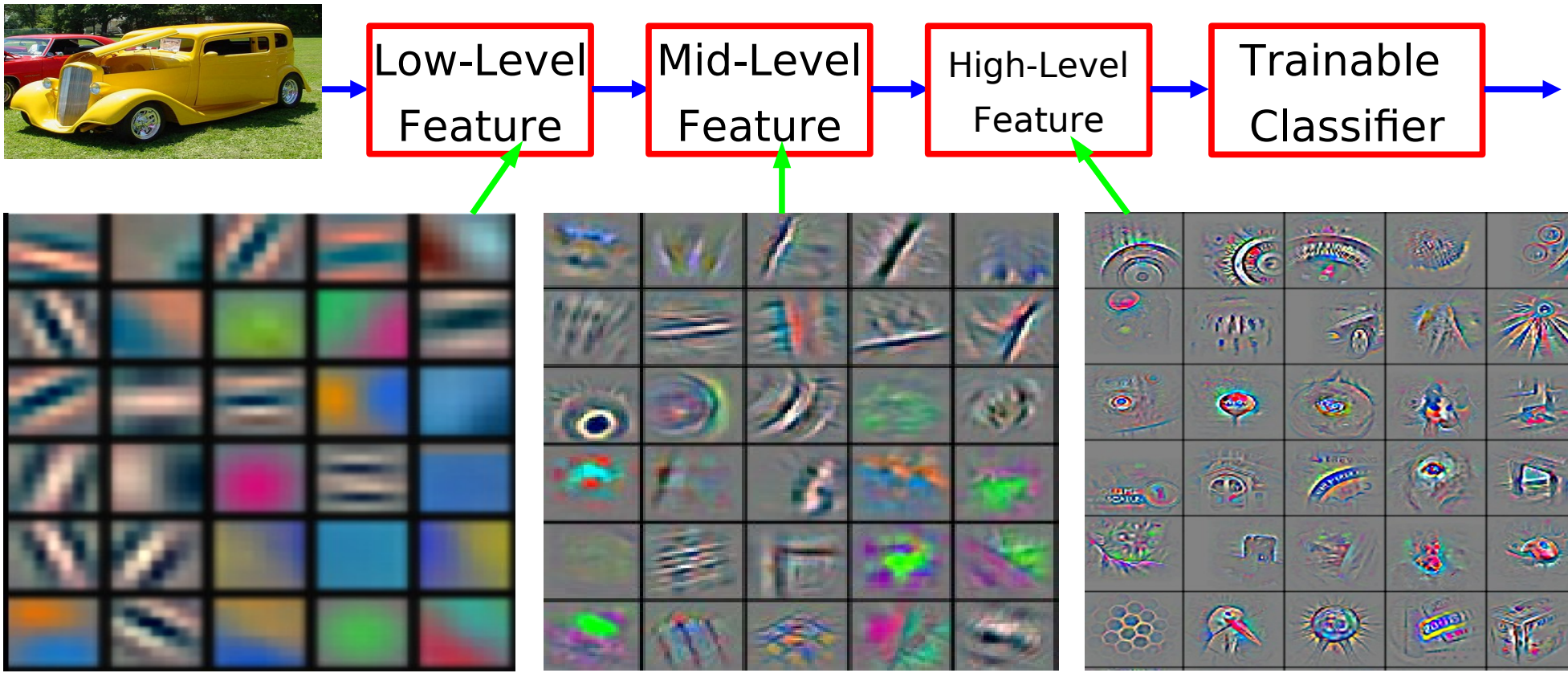
GOPS vs Accuracy on ImageNet vs #Parameters

- **[Canziani 2016]**
- **ResNet50 and ResNet100 are used routinely in production.**
- **Each of the few billions photos uploaded on Facebook every day goes through a handful of ConvNets within 2 seconds.**



Multilayer Architectures == Compositional Structure of Data

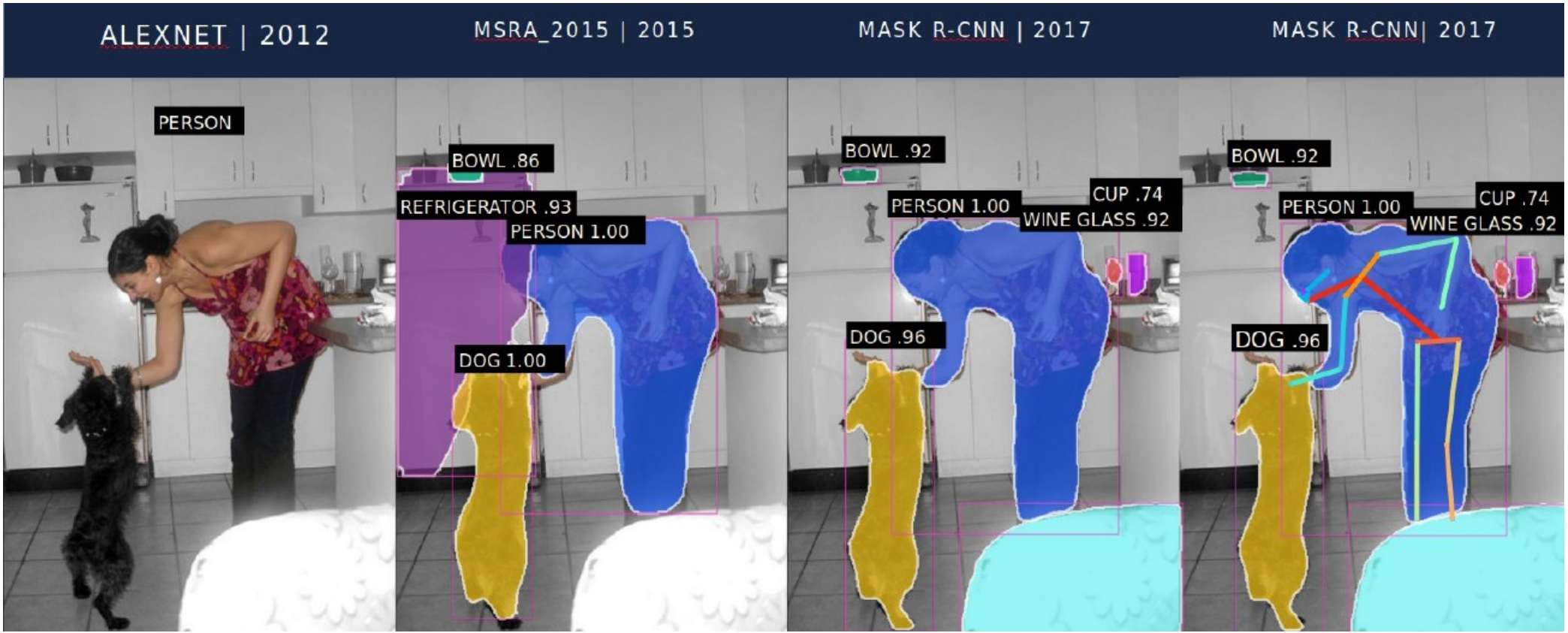
- **Natural is data is compositional => it is efficiently representable hierarchically**



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

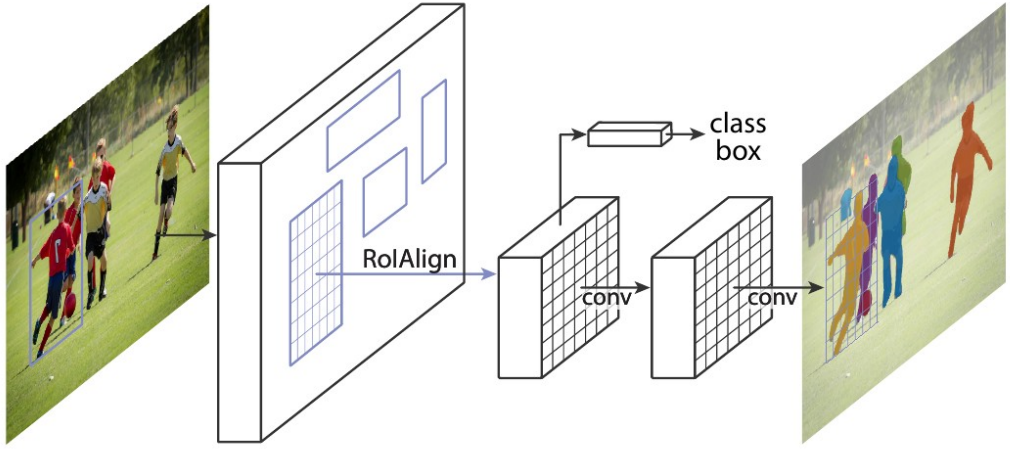
Progress in Computer Vision

- **[He 2017]**

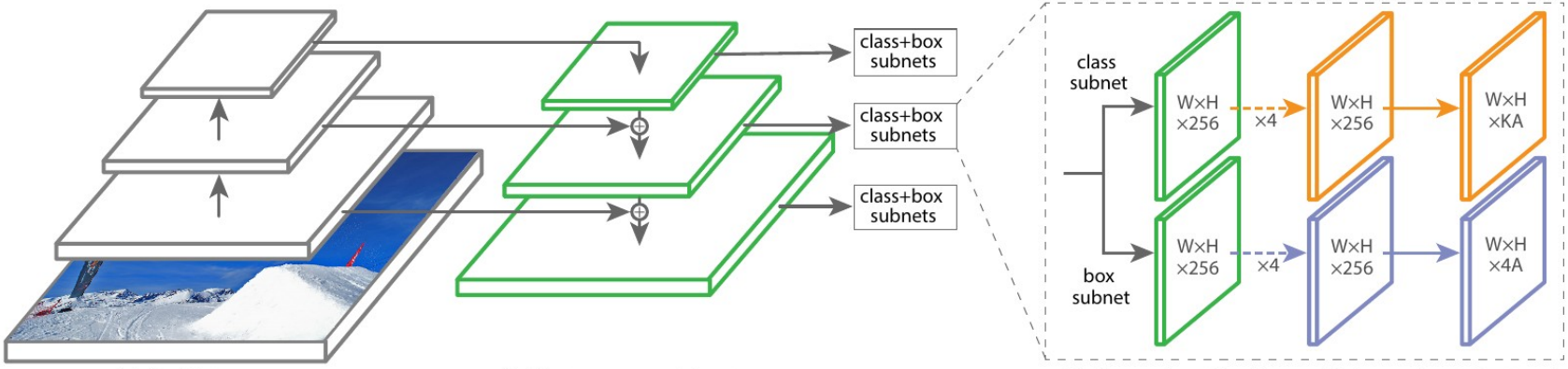


Mask-RCNN, RetinaNet, feature pyramid network

- **Mask-RCNN**
  - [He et al. arXiv:1703.06870]
  - ConvNet produces an object mask for each region of interest



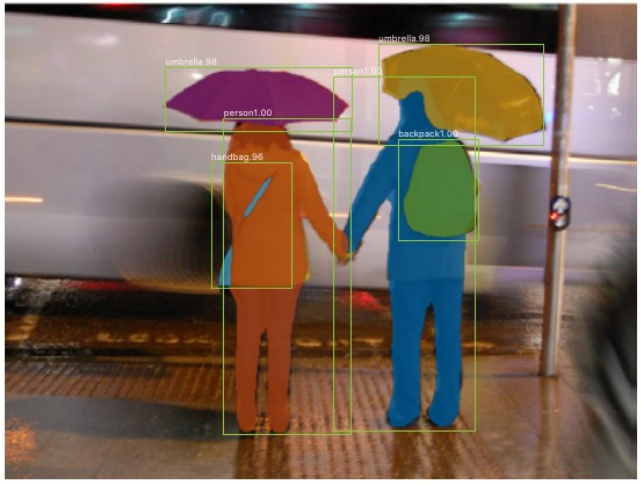
- **RetinaNet/FPN**
  - [Lin et al. ArXiv:1708.02002]
  - one-pass object detection

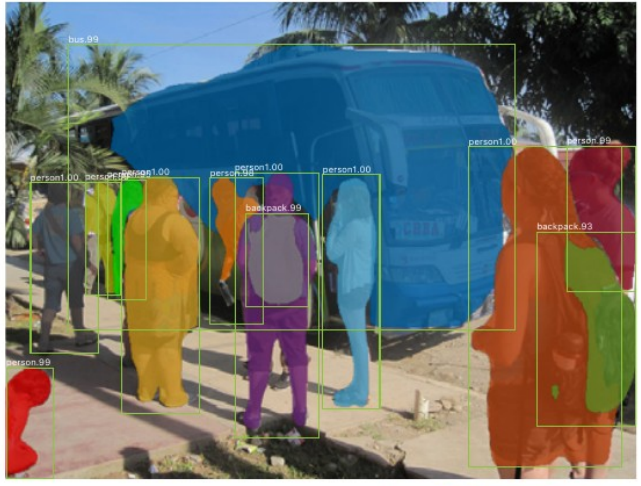


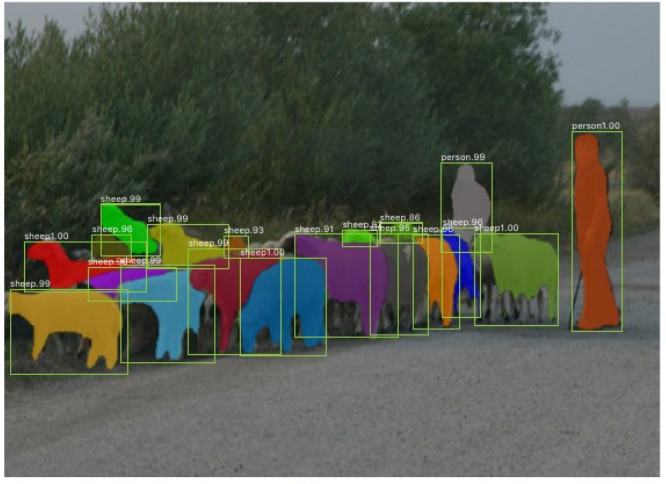
Mask-RCNN Results on COCO dataset

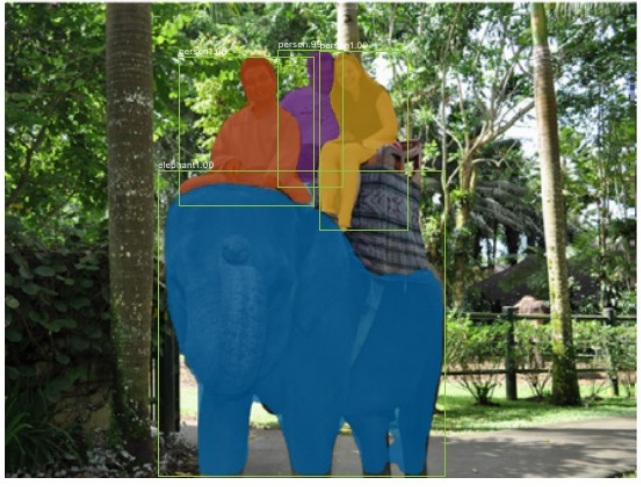
- **Individual objects are segmented.**

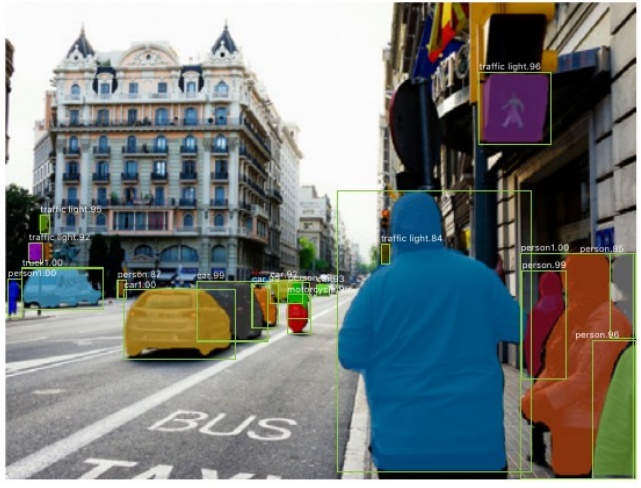
Mask R-CNN Results on COCO test set

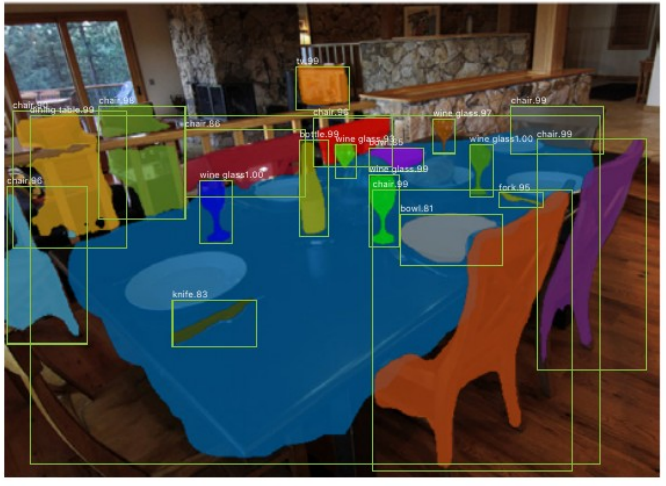






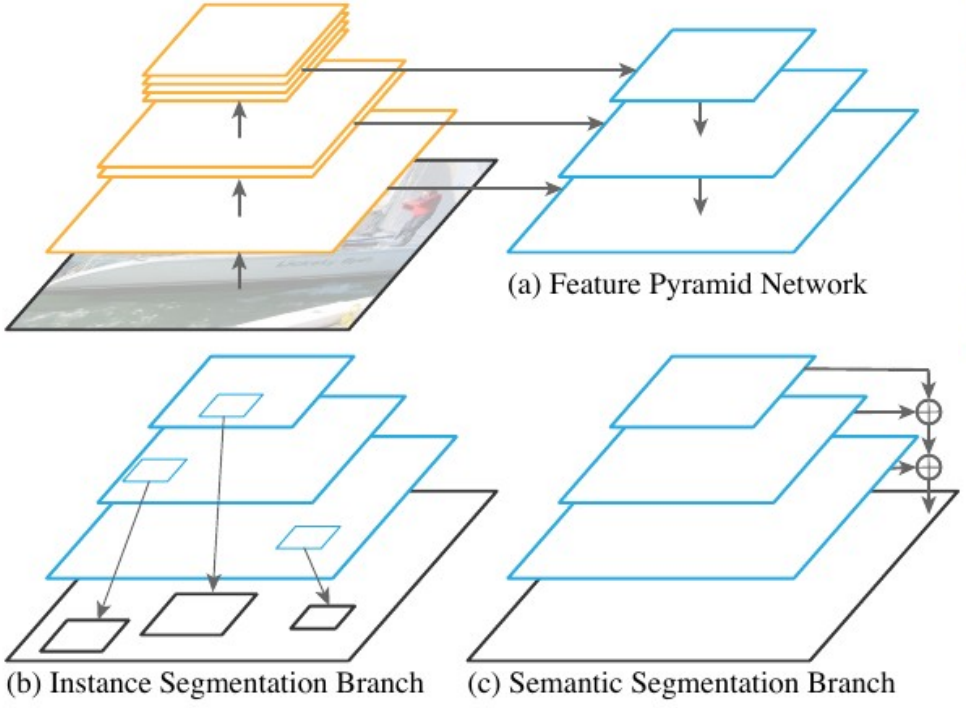






Panoptic Feature Pyramid Network

- **Segments and recognizes object instances and regions**
  - [Kirillov arXiv:1901.0244]

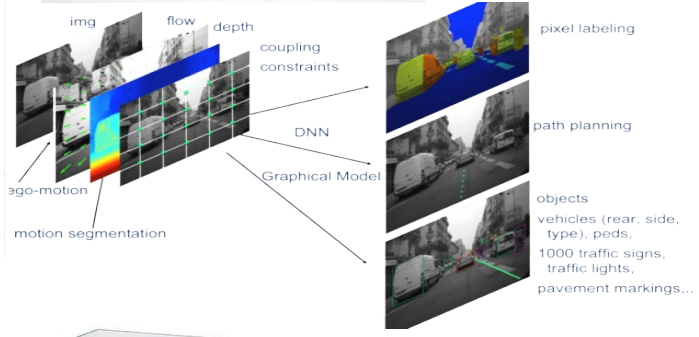


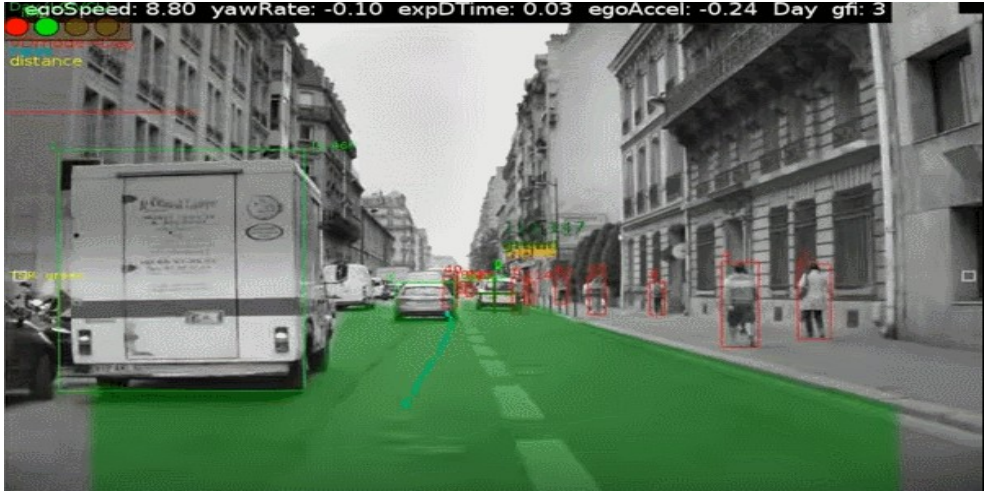
Detectron2 (FAIR) [Girshick 2019]

- **Panoptic instance segmentation, (dense) body pose estimation**
- **Open source: https://github.com/facebookresearch/detectron2**

Driving Cars with Convolutional Nets

- **MobilEye (2015)**

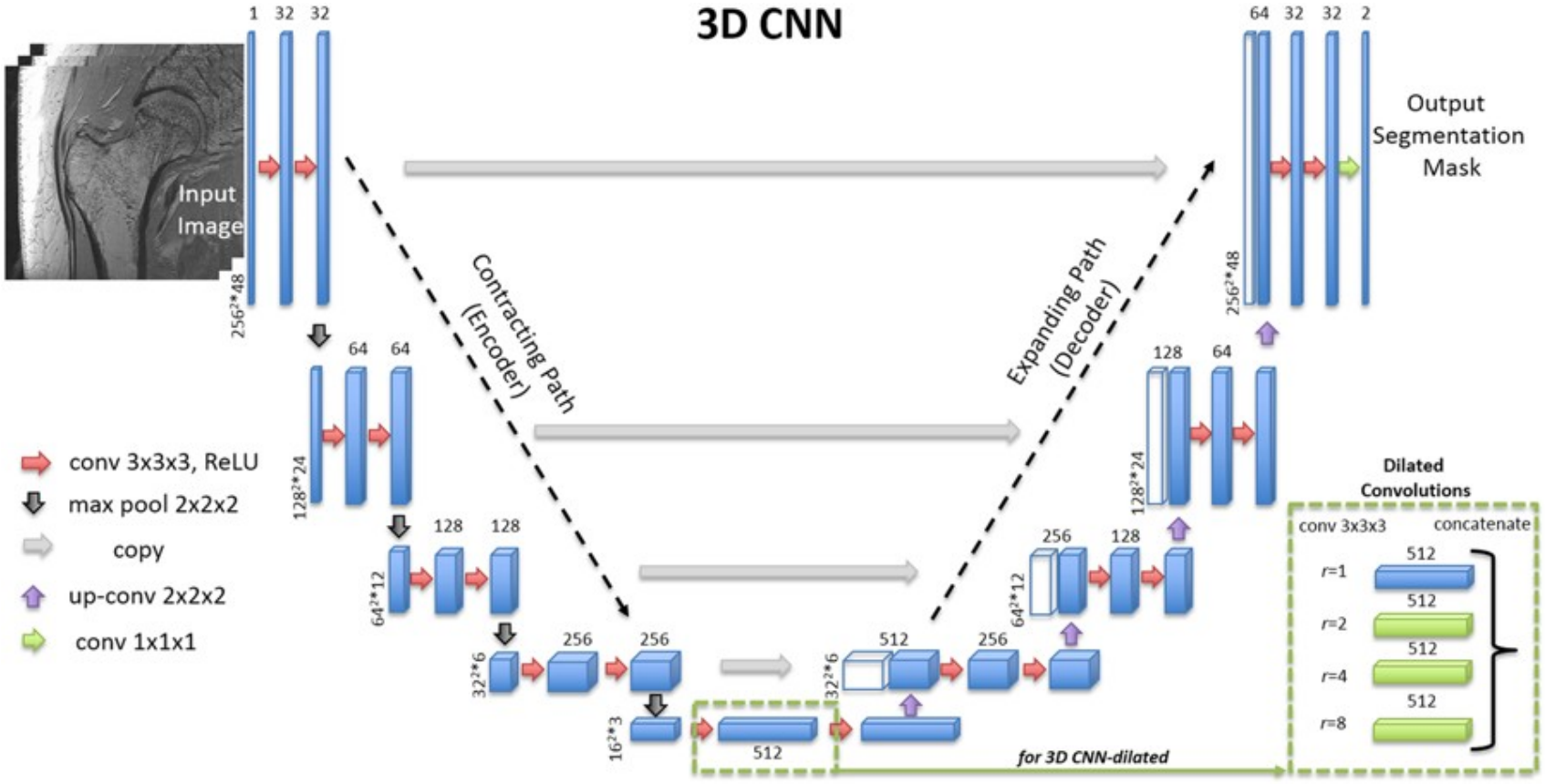




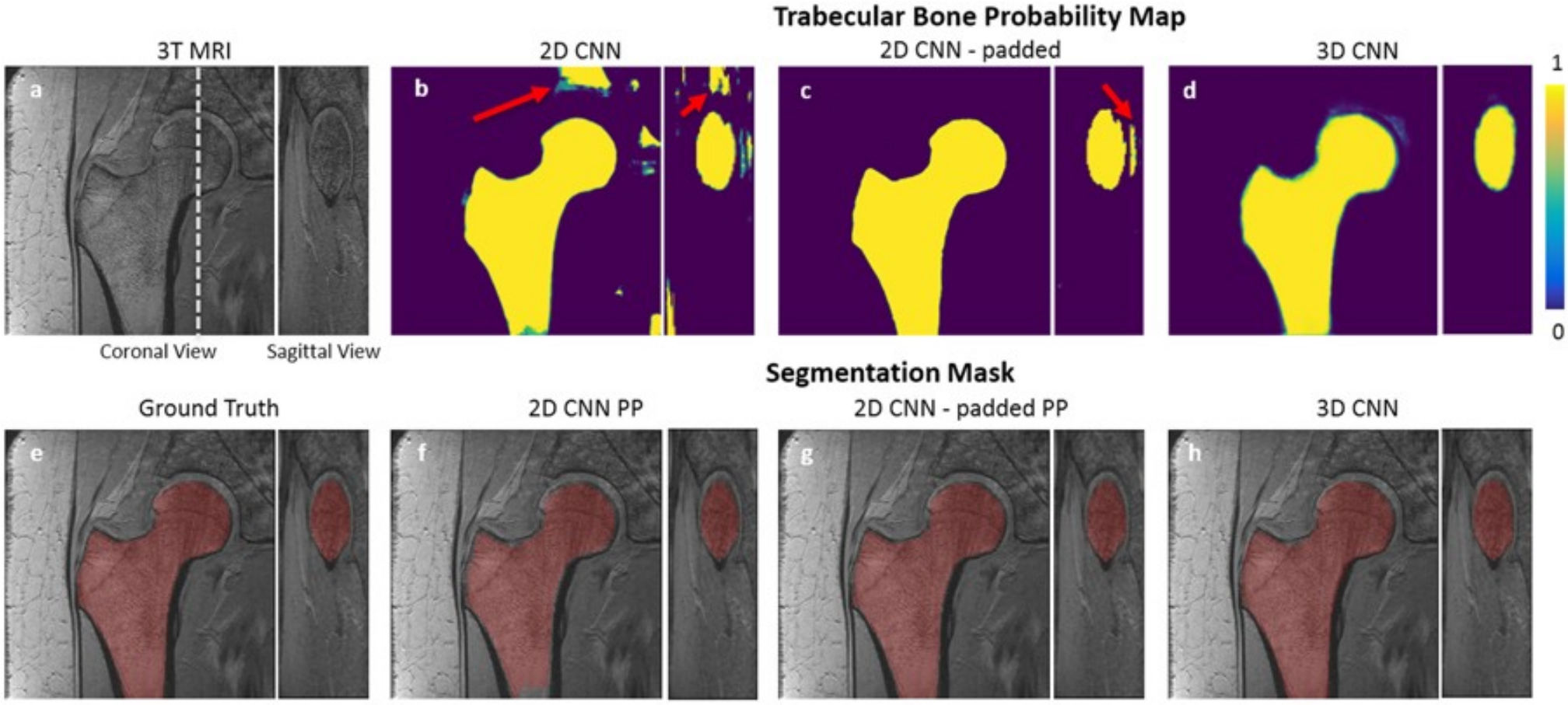
- **NVIDIA**

3D ConvNet for Medical Image Analysis (NYU)

- **Segmentation Femur from MR Images**
- **[Deniz et al. Nature 2018]**

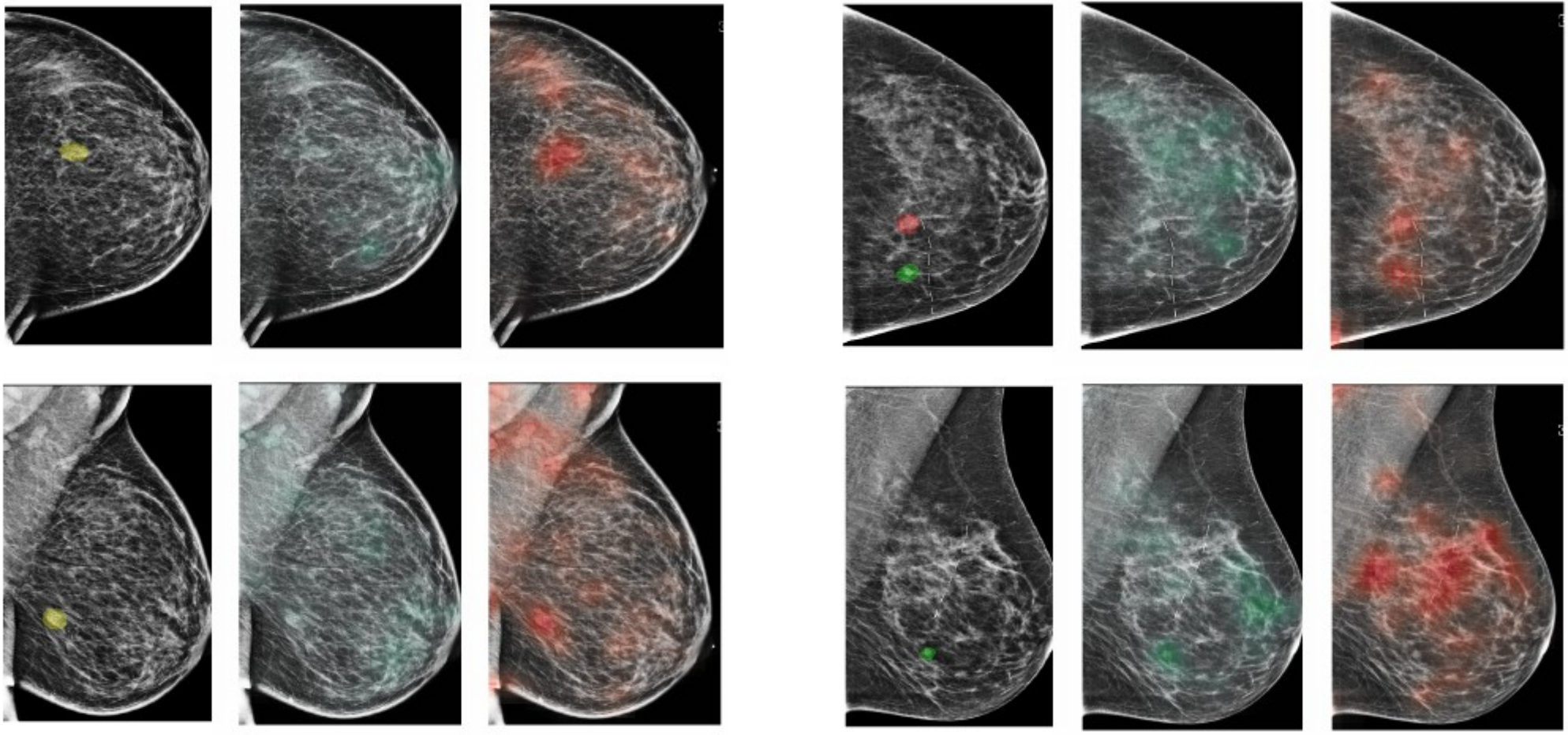


3D ConvNet for Medical Image Analysis (NYU)



Breast Cancer Detection (NYU)

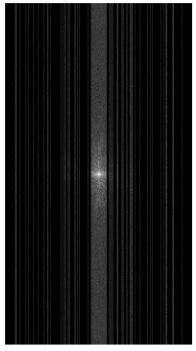
- **[Wu et al. ArXiv:1903.08297] https://github.com/nyukat/breast_cancer_classifier**



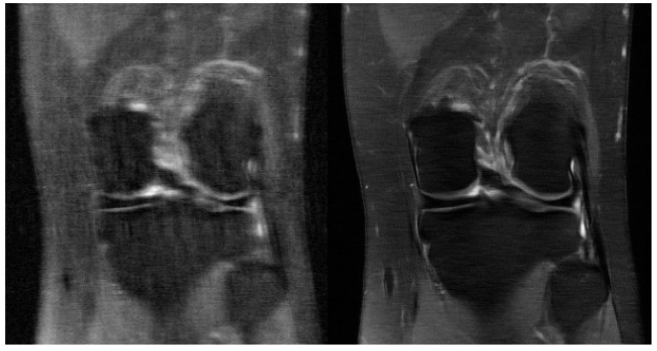
FastMRI (NYU+FAIR): 4x-8x speed up for MRI data acquisition

- MRI images subsampled (in k-space) by 4x and 8x
- [Zbontar et al. ArXiv:1811.08839]
- U-Net architecture
- 4-fold acceleration
- 8-fold acceleration
- K-space masks

(a) Cropped and vertically flipped reconstruction from fully sampled k-space data

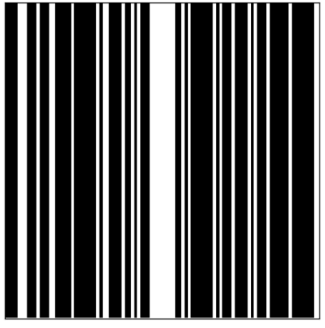


(b) Rectangular masked k-space

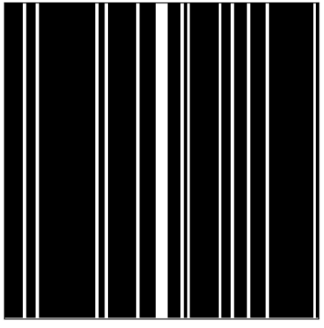


(c) Reconstruction via zero-filled IFFT

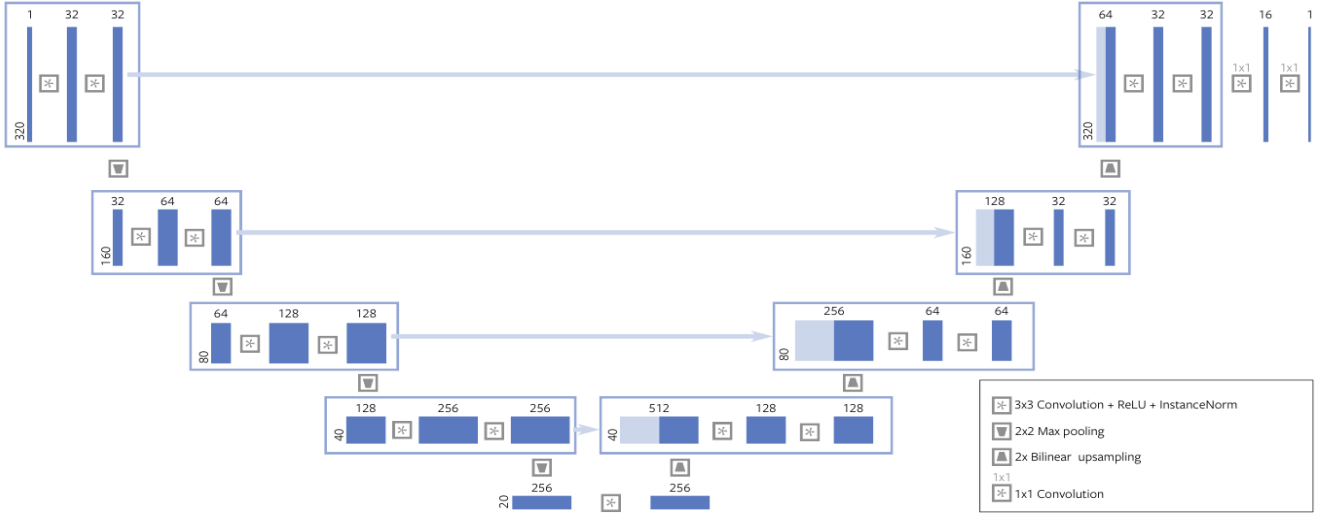
(d) Deep-learning baseline UNET reconstruction



(a) 4-fold acceleration



(b) 8-fold acceleration



ConvNets (and Deep Learning) in Physics

- **Approximate solutions of PDEs with a learned update**
  - Integration step of PDE solver: Z(t+1) = Z(t) + dt*G(Z(t))

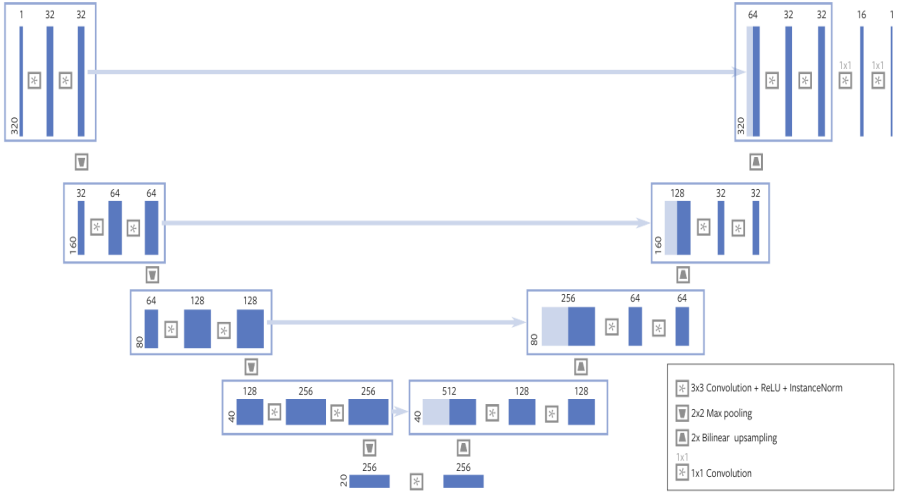
    where is G() a translation-invariant local operator.

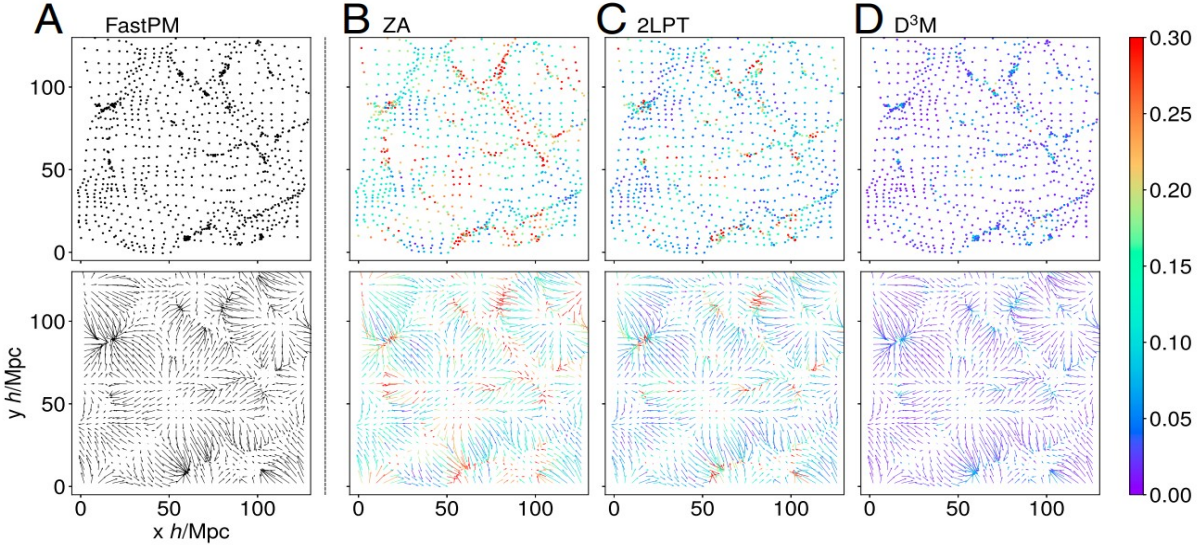
    Example: G(Z(t)) = V*f(W*Z(t)) conv->transfer_func->conv
- **High energy Physics**
  - Lattice QCD
- **Fluid Dynamics**
  - Prediction of aero/hydro-dynamical properties of solids
  - Shape refinement by gradient descent
- **Cosmology / Astrophysics**
  - Large-scale simulation of the early universe

ConvNets in Astrophysics [He et al. PNAS 07/2019]

Learning to predict the cosmological structure formation

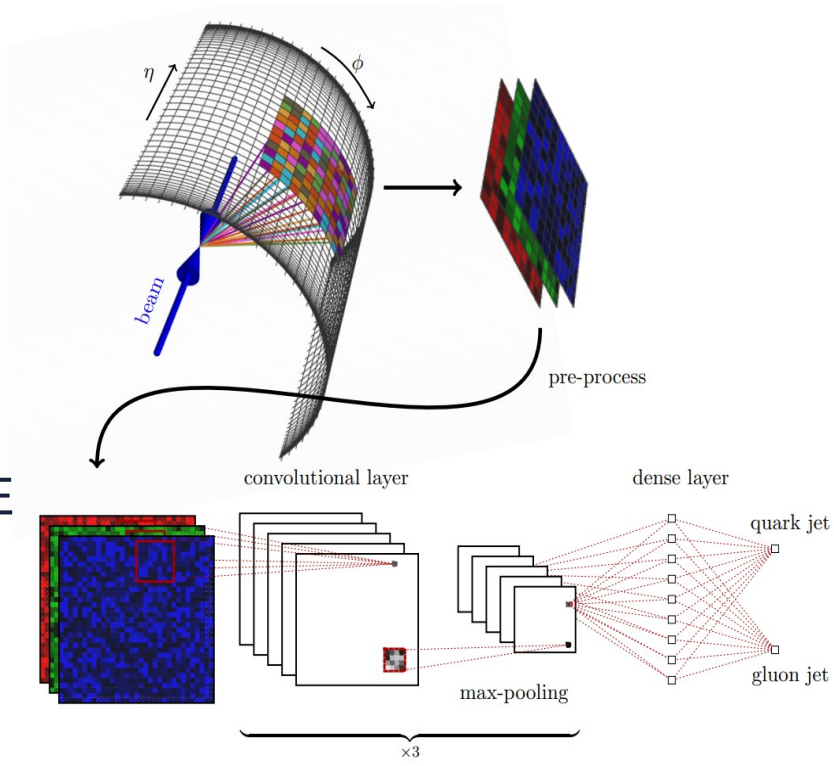
- 1. Train a coarse-grained 3D U-Net to approximate a fine-grained simulation on a small volume
- 2. Use it for a simulation on a large volume (the early universe)





ConvNets (and Deep Learning) in Physics

- **Material Science / Molecular dynamics**
  - Protein structure/function prediction
  - Prediction of material properties
- **High energy Physics**
  - Jet filtering / analysis
    - "Deep learning in color: towards automated quark/gluon jet discrimination", P Komiske, E Metodiev, M Schwartz, arXiv:1612.01551
- **Cosmology / Astrophysics**
  - Infering constants from observations
  - Statistical studies of galaxies,
  - Dark matter through gravitational lensing

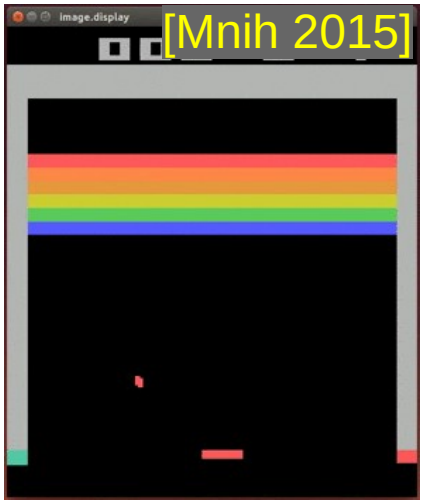


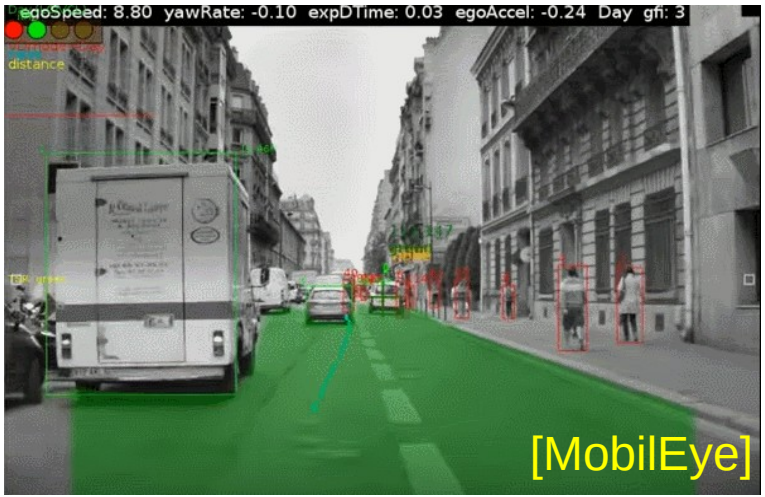
Applications of ConvNets

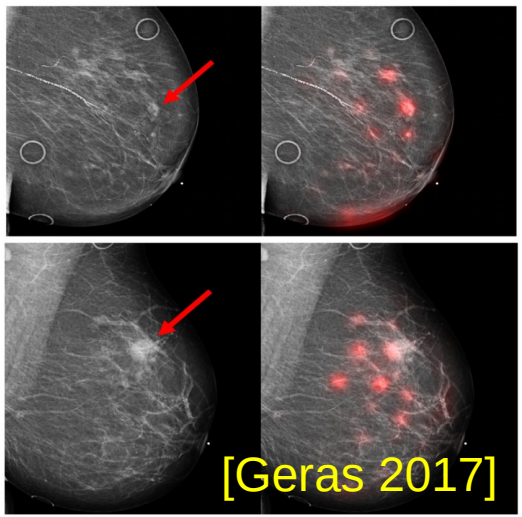
- **Self-driving cars, visual perception**
- **Medical signal and image analysis**
  - Radiology, dermatology, EEG/seizure prediction….
- **Bioinformatics/genomics**
- **Speech recognition**
- **Language translation**
- **Image restoration/manipulation/style transfer**
- **Robotics, manipulation**
- **Physics**
  - High-energy physics, astrophysics
- **New applications appear every day**
  - E.g. environmental protection,….

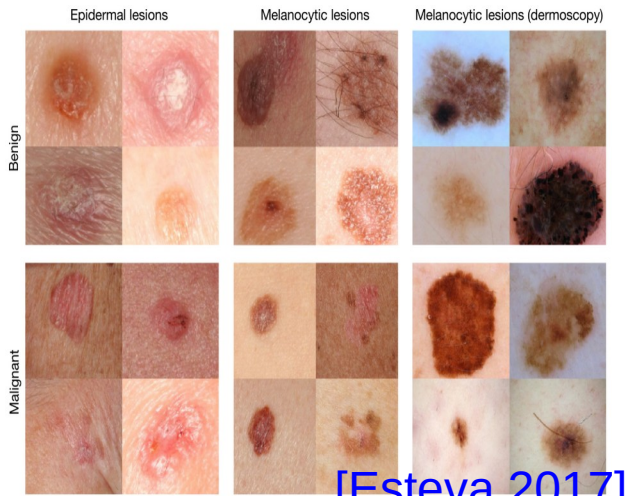
Applications of Deep Learning

- **Medical image analysis**
- **Self-driving cars**
- **Accessibility**
- **Face recognition**
- **Language translation**
- **Virtual assistants***
- **Content Understanding for:**
  - Filtering
  - Selection/ranking
  - Search
- **Games**
- **Security, anomaly detection**
- **Diagnosis, prediction**
- **Science!**

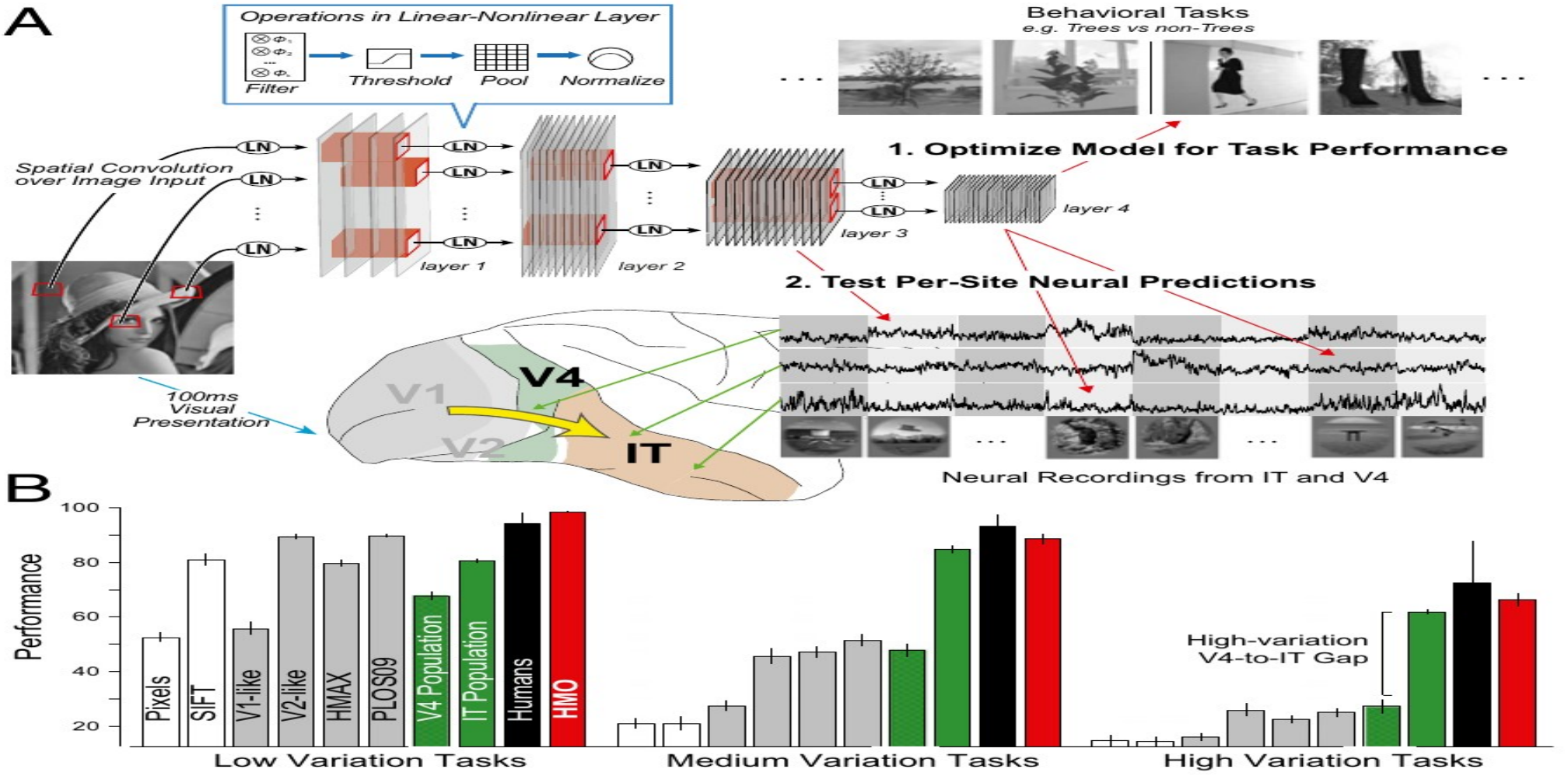








ConvNets & The Visual System [Yamins et al. PNAS 2014]



ConvNets as Models of the Visual System?

- **[Yamins & Di Carlo 2016]**

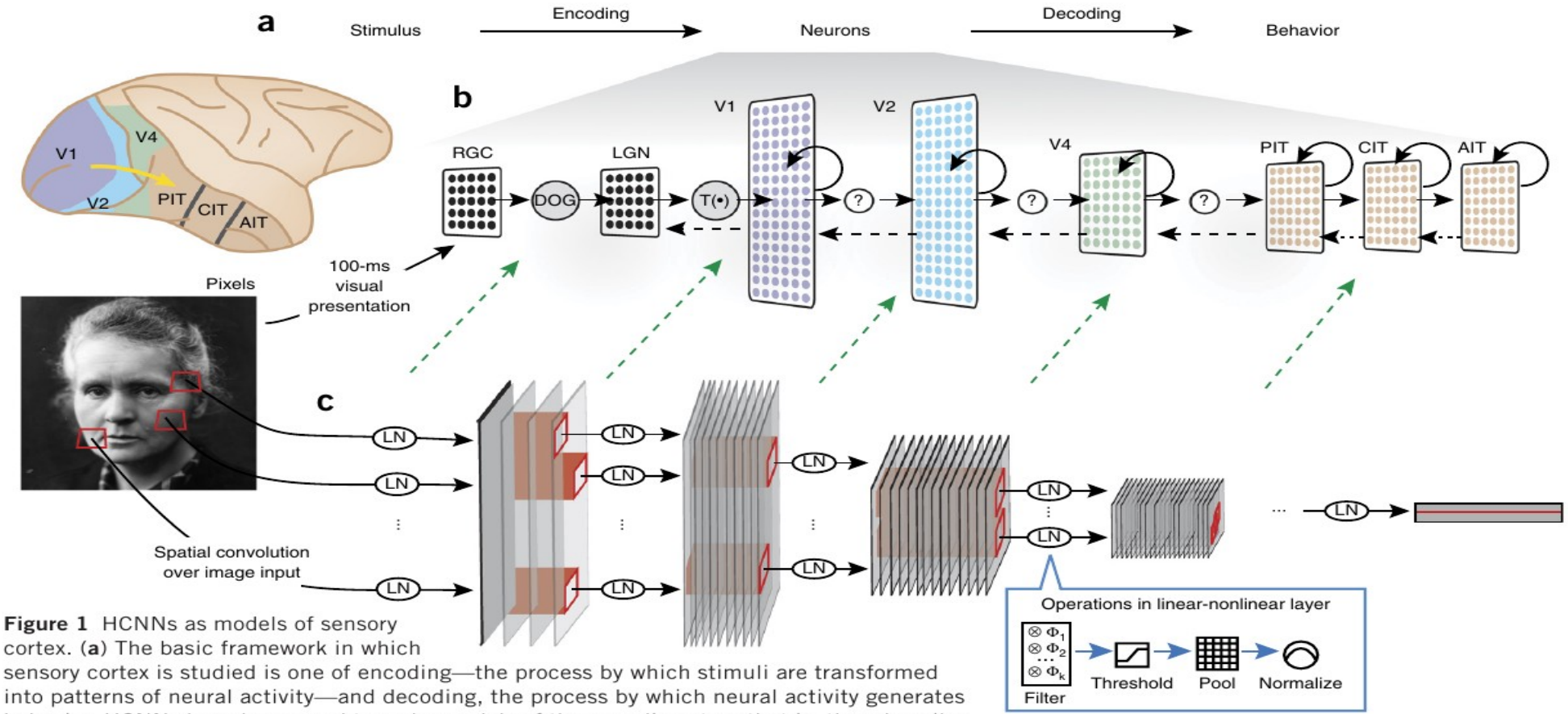
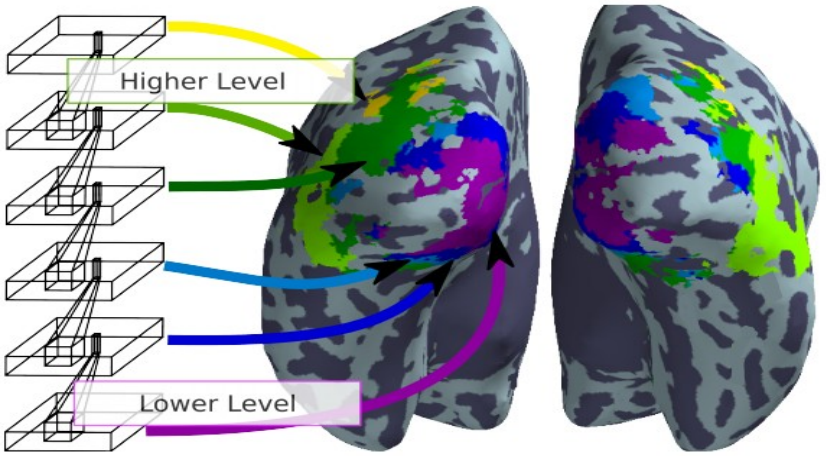


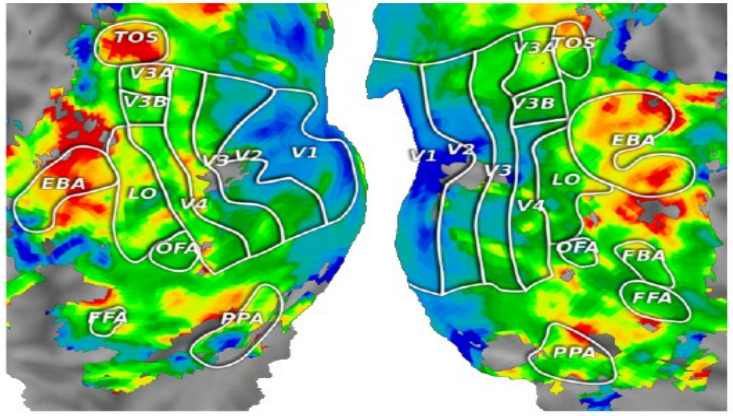
Figure 1 HCNNs as models of sensory cortex. (**a**) The basic framework in which sensory cortex is studied is one of encoding—the process by which stimuli are transformed into patterns of neural activity—and decoding, the process by which neural activity generates behavior. HCNNs have been used to make models of the encoding step; that is, they describe

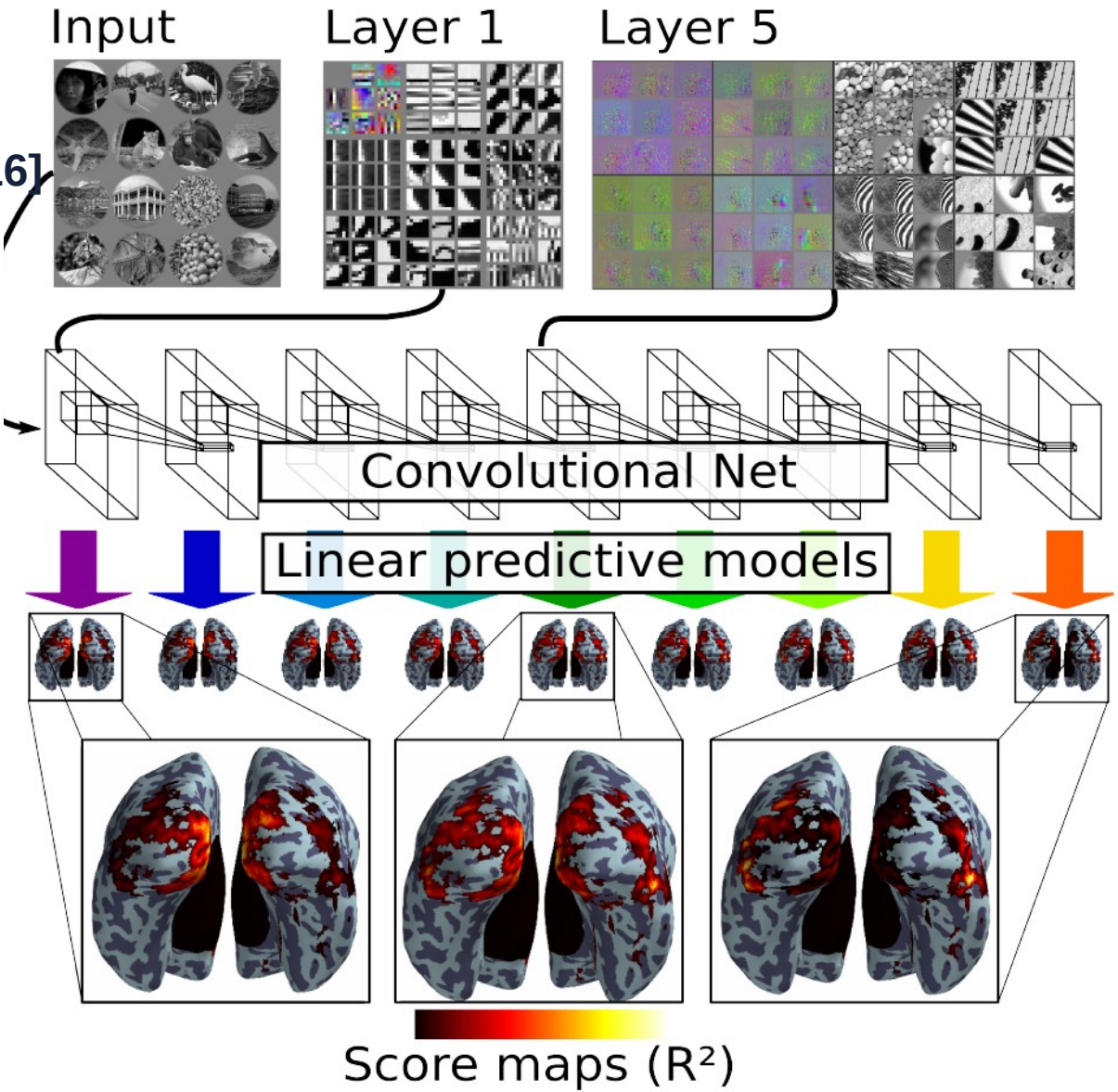
ConvNet models & fMRI

- **[Eickenberg et al. NeuroImage 2016]**



B Fingerprint summaries for Huth2012

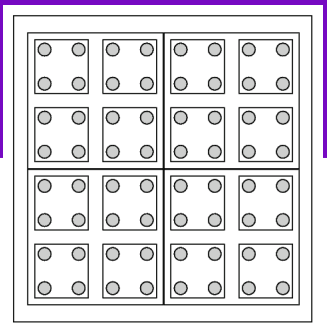




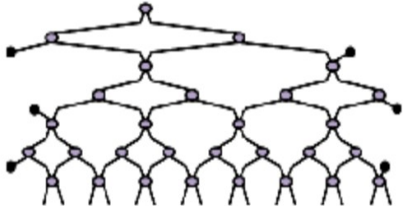
Why does it work so well?

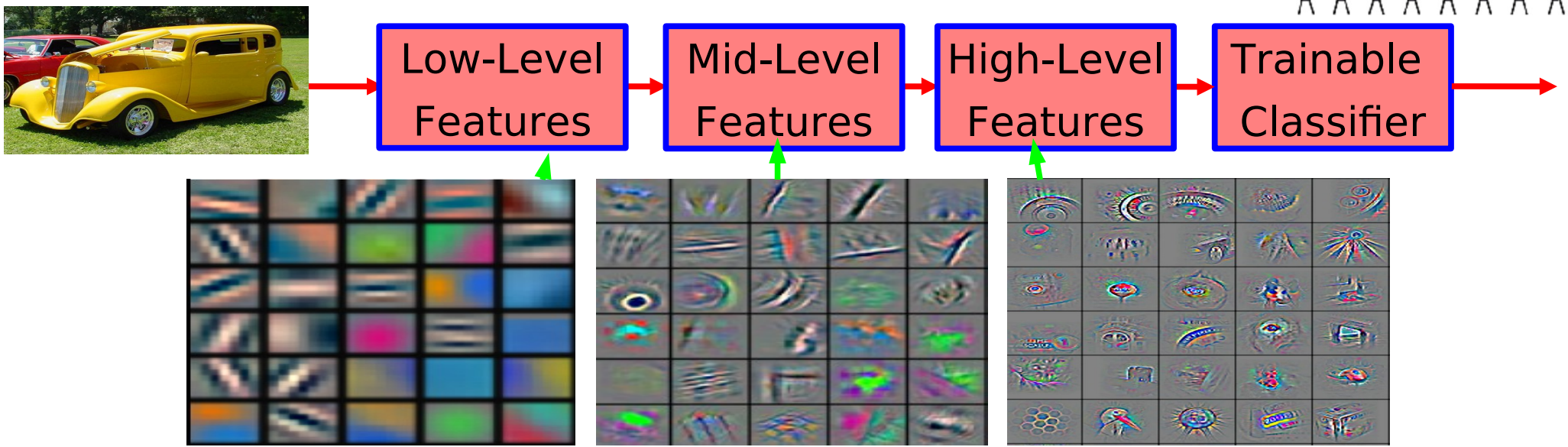
- **We can approximate any function with two layers**
  - Why do we need layers?
- **What is so special convolutional networks?**
  - Why do they work so well on natural signals?
- **The objective function are highly non-convex.**
  - Why doesn’t SGD get trapped in local minima?
- **The networks are widely over-parameterized.**
  - Why do they not overfit?

The world is compositional



- **Convolutional networks learn hierarchical representations**
  - Upper-layer representation are at a coarse spatial scale
    - Renormalization group theory
    - Multi-scale entanglement renormalization ansatz (MERA)





Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

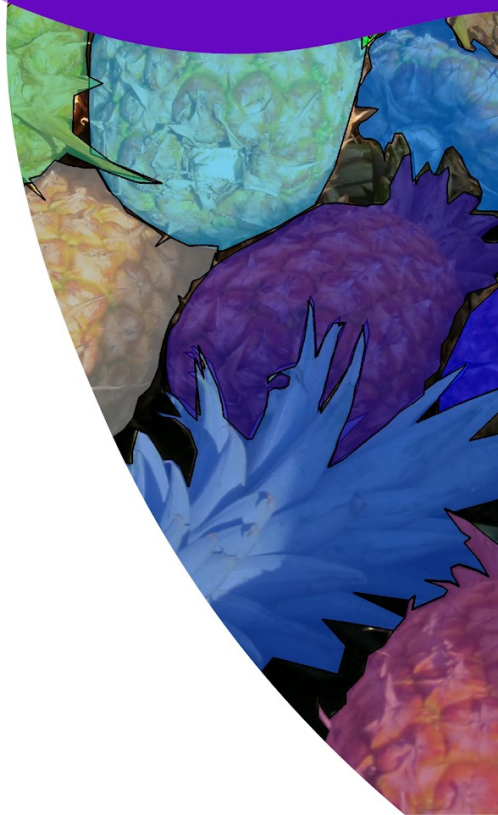
What current deep learning methods enables

- **What we can have**
  - Safer cars, autonomous cars
  - Better medical image analysis
  - Personalized medicine
  - Adequate language translation
  - Useful but stupid chatbots
  - Information search, retrieval, filtering
  - Numerous applications in energy, finance, manufacturing, environmental protection, commerce, law, artistic creation, games,…..
- **What we cannot have (yet)**
  - Machines with common sense
  - Intelligent personal assistants
  - “Smart” chatbots”
  - Household robots
  - Agile and dexterous robots
  - Artificial General Intelligence (AGI)

NEW YORK UNIVERSITY

Learning Representations

What are good representations?
Why do networks need to be deep?



Deep Learning = Learning Representations/Features

- **The traditional model of pattern recognition (since the late 50's)**
  - Fixed/engineered features (or fixed kernel) + trainable classifier

hand-crafted Feature Extractor → "Simple" Trainable Classifier →

- **End-to-end learning / Feature learning / Deep learning**
  - Trainable features (or kernel) + trainable classifier

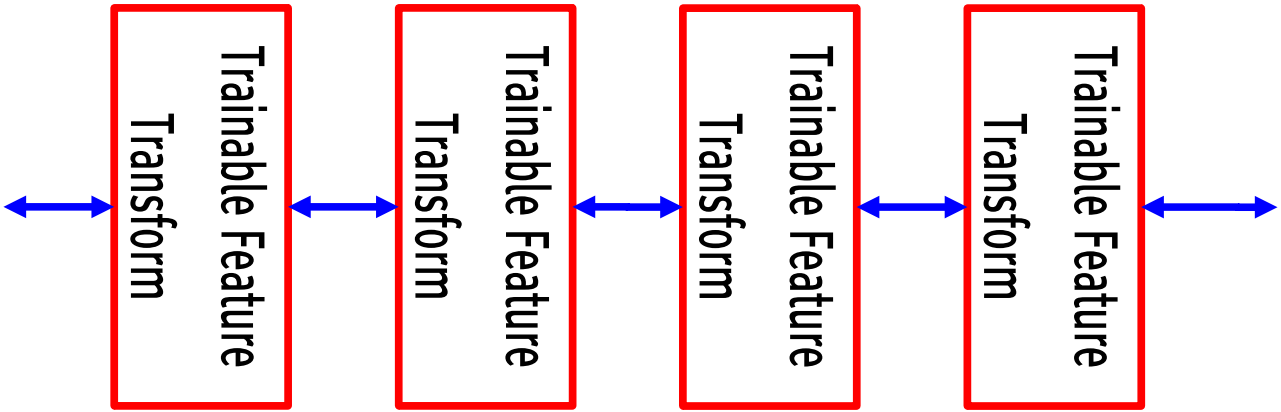
Trainable Feature Extractor → Trainable Classifier →

Ideas for “generic” feature extraction

- **Basic principle:**
  - expanding the dimension of the representation so that things are more likely to become linearly separable.
-
- **- space tiling**
- **- random projections**
- **- polynomial classifier (feature cross-products)**
- **- radial basis functions**
- **- kernel machines**

Hierarchical representation

- **Hierarchy of representations with increasing level of abstraction**
- **Each stage is a kind of trainable feature transform**
- **Image recognition**
  - Pixel → edge → texton → motif → part → object
- **Text**
  - Character → word → word group → clause → sentence → story
- **Speech**
  - Sample → spectral band → sound → … → phone → phoneme → word



Do we really need deep architectures?

- **Theoretician's dilemma: "We can approximate any function as close as we want with shallow architecture. Why would we need deep ones?"**

$$y = \sum_{i=1}^{P} \alpha_i K(X, X^i) \qquad y = F(W^1.F(W^0.X))$$

  - kernel machines (and 2-layer neural nets) are "universal".

- **Deep learning machines**

$$y = F(W^K.F(W^{K-1}.F(....F(W^0.X)...)))$$

- **Deep machines are more efficient for representing certain classes of functions, particularly those involved in visual recognition**
  - they can represent more complex functions with less "hardware"
- **We need an efficient parameterization of the class of functions that are useful for "AI" tasks (vision, audition, NLP...)**

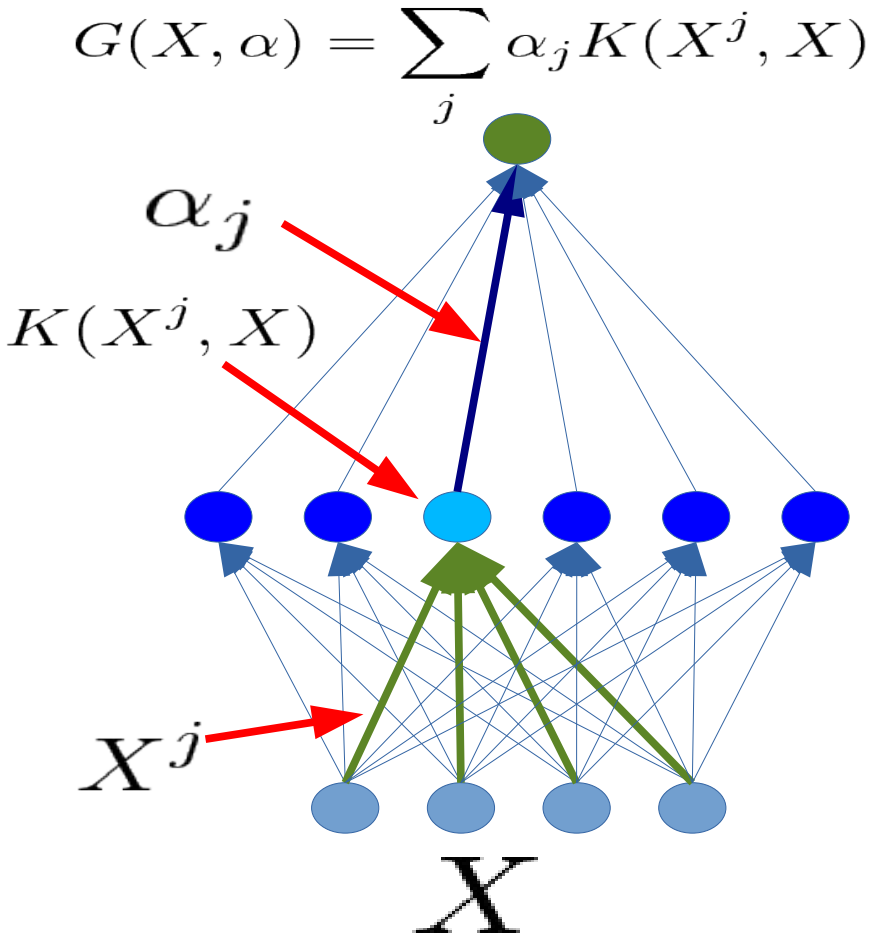
Why would deep architectures be more efficient?

[Bengio & LeCun 2007 “Scaling Learning Algorithms Towards AI”]

- **A deep architecture trades space for time (or breadth for depth)**
  - more layers (more sequential computation),
  - but less hardware (less parallel computation).
- **Example1: N-bit parity**
  - requires N-1 XOR gates in a tree of depth log(N).
  - Even easier if we use threshold gates
  - requires an exponential number of gates of we restrict ourselves to 2 layers (DNF formula with exponential number of minterms).
- **Example2: circuit for addition of 2 N-bit binary numbers**
  - Requires O(N) gates, and O(N) layers using N one-bit adders with ripple carry propagation.
  - Requires lots of gates (some polynomial in N) if we restrict ourselves to two layers (e.g. Disjunctive Normal Form).
  - Bad news: almost all boolean functions have a DNF formula with an exponential number of minterms O(2^N).....

Which Models are Deep?

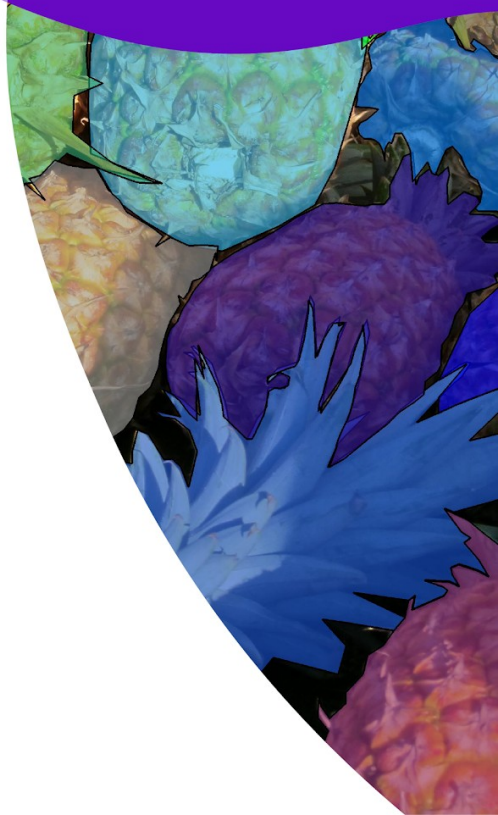
- **2-layer models are not deep (even if you train the first layer)**
  - Because there is no feature hierarchy
- **Neural nets with 1 hidden layer are not deep**
- **SVMs and Kernel methods are not deep**
  - Layer1: kernels; layer2: linear
  - The first layer is “trained” in with the simplest unsupervised method ever devised: using the samples as templates for the kernel functions.
- **Classification trees are not deep**
  - No hierarchy of features. All decisions are made in the input space



NEW YORK UNIVERSITY

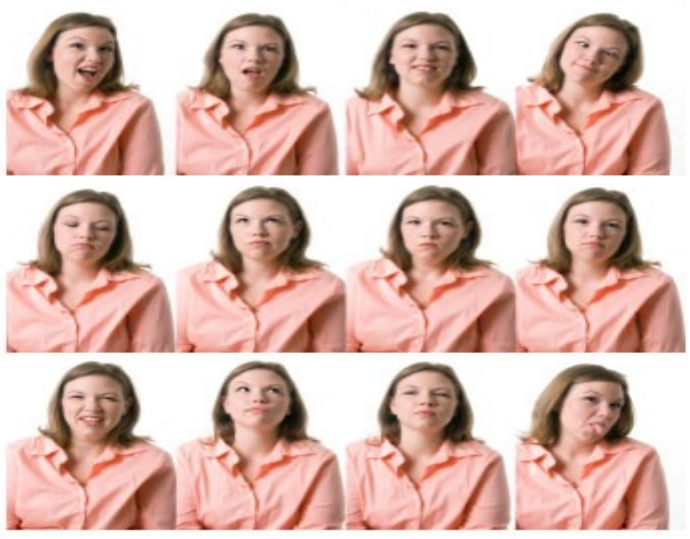
What are Good Features?

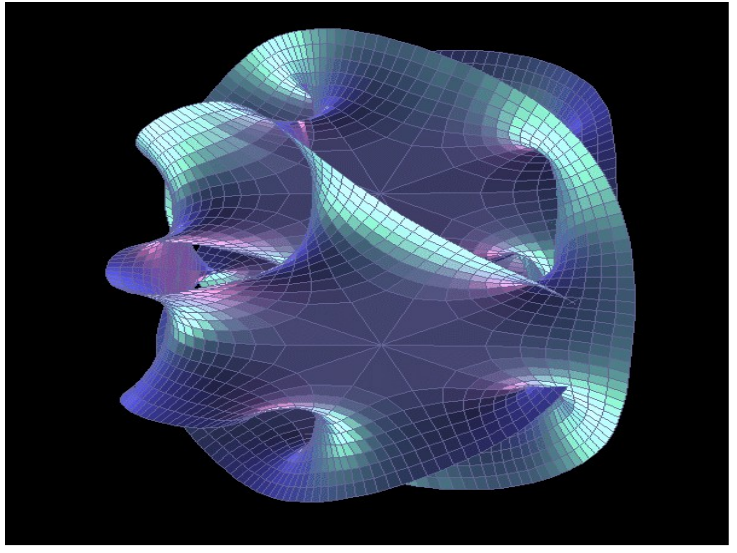
What are good representations?



Discovering the Hidden Structure in High-Dimensional Data: The manifold hypothesis

- **Learning Representations of Data:**
  - **Discovering & disentangling the independent explanatory factors**
- **The Manifold Hypothesis:**
  - Natural data lives in a low-dimensional (non-linear) manifold
  - Because variables in natural data are mutually dependent





Discovering the Hidden Structure in High-Dimensional Data

- **Example: all face images of a person**
  - 1000x1000 pixels = 1,000,000 dimensions
  - But the face has 3 Cartesian coordinates and 3 Euler angles
  - And humans have less than about 50 muscles in the face
  - Hence the manifold of face images for a person has <56 dimensions
- **The perfect representations of a face image:**
  - Its coordinates on the face manifold
  - Its coordinates away from the manifold
- **We do not have good and general methods to learn functions that turns an image into this kind of representation**

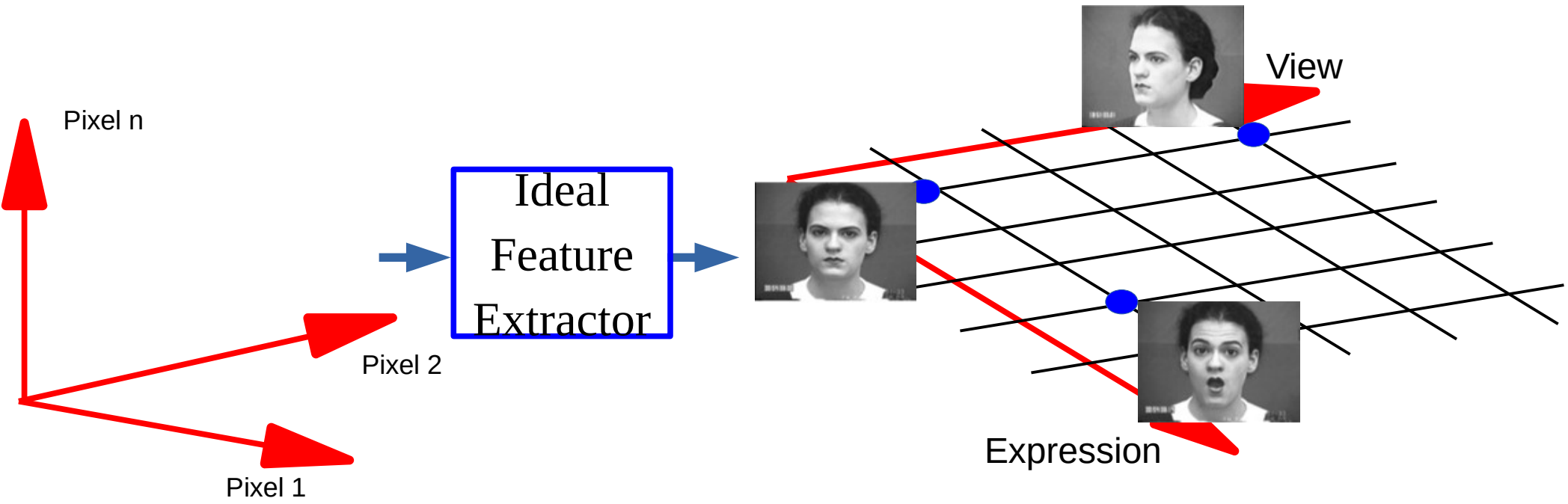
Ideal Feature Extractor

$$\begin{bmatrix} 1.2 \\ -3 \\ 0.2 \\ -2 \dots \end{bmatrix}$$

Face/not face
Pose
Lighting
Expression

Disentangling factors of variation

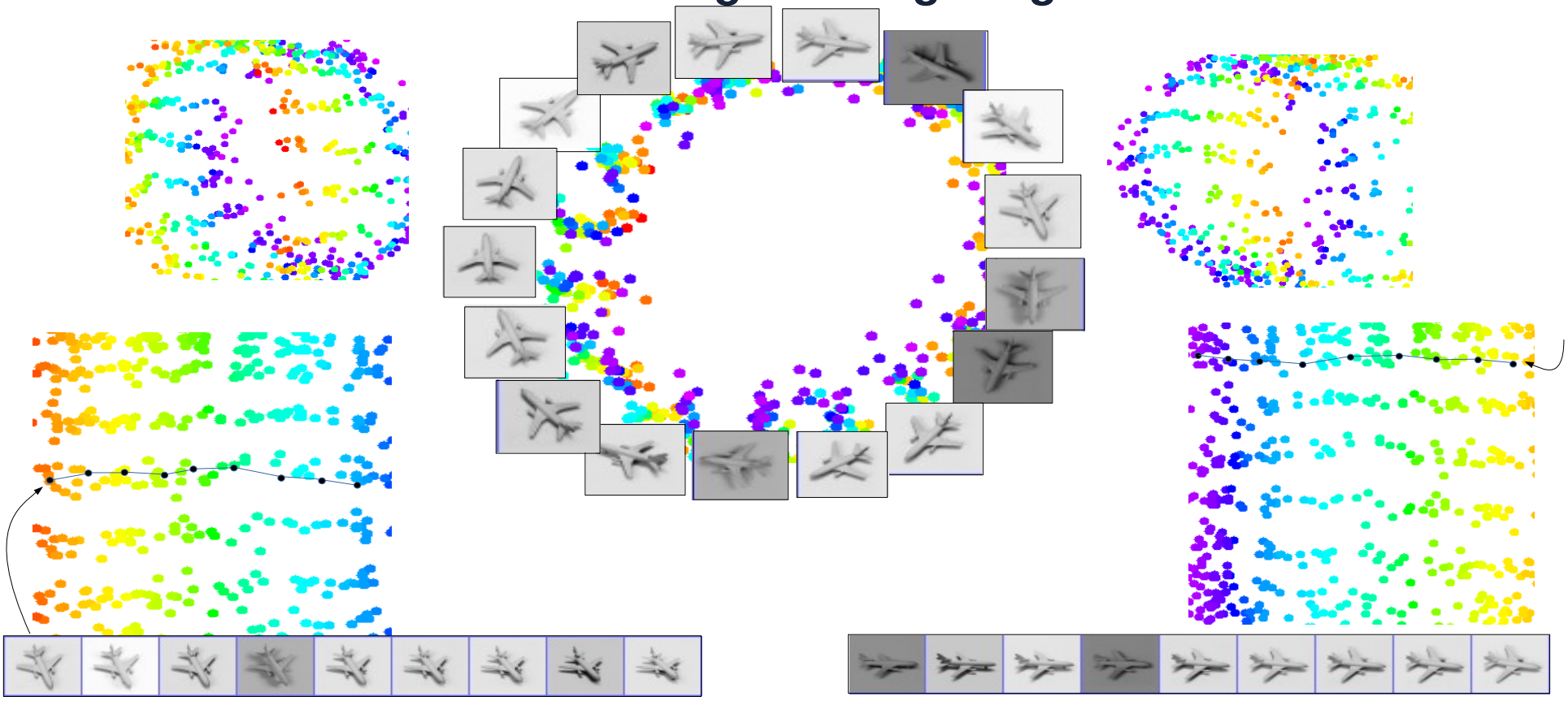
The Ideal Disentangling Feature Extractor



Data Manifold & Invariance: Some variations must be eliminated

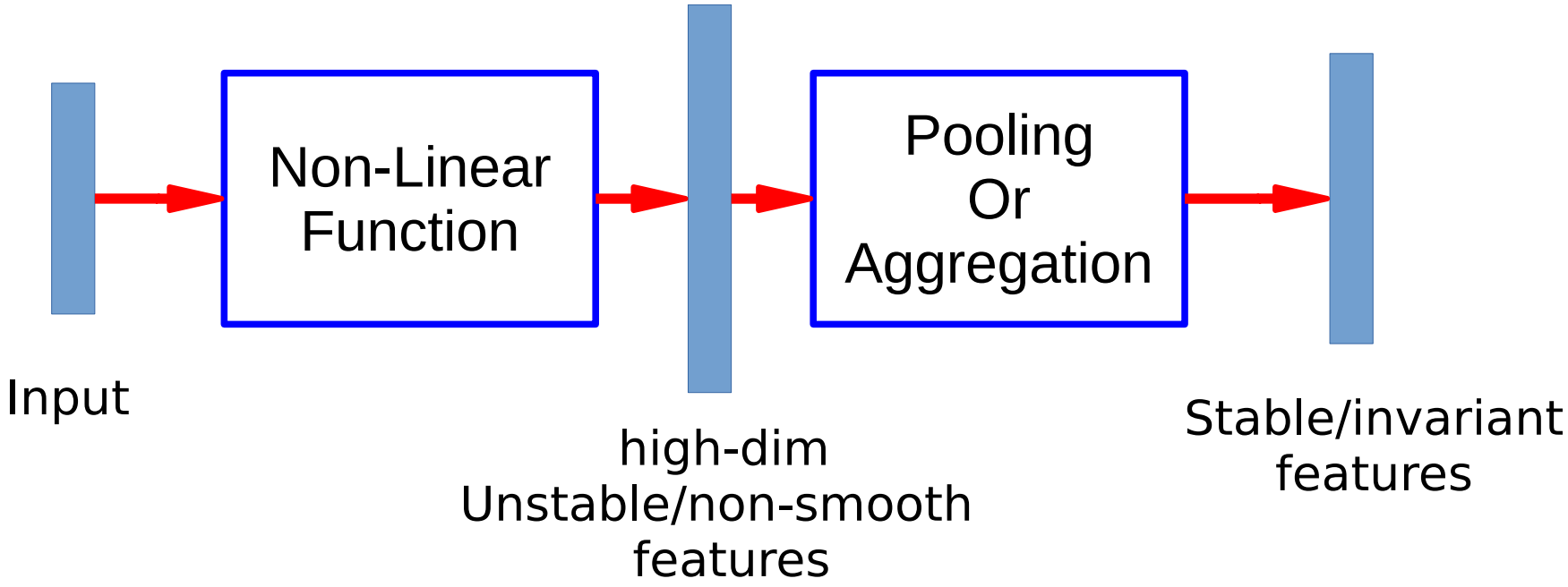
- **Azimuth-Elevation manifold. Ignores lighting.**

[Hadsell et al. CVPR 2006]



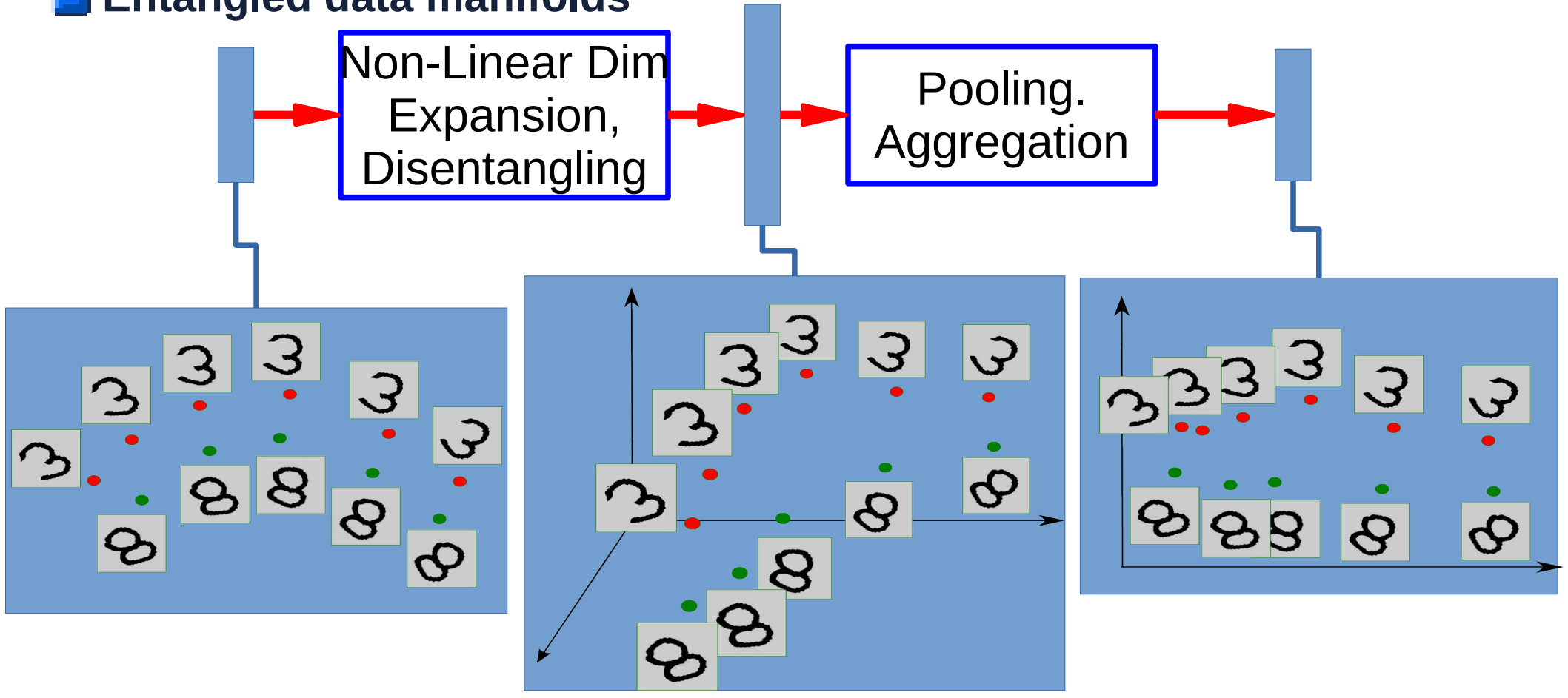
Basic Idea for Invariant Feature Learning

- **Embed the input non-linearly into a high(er) dimensional space**
  - In the new space, things that were non separable may become separable
- **Pool regions of the new space together**
  - Bringing together things that are semantically similar. Like pooling.



Non-Linear Expansion → Pooling

- **Entangled data manifolds**



Sparse Non-Linear Expansion → Pooling

- **Use clustering to break things apart, pool together similar things**